Mathematics *Standards of Learning* for Virginia Public Schools 2023 Computer Mathematics

This course is intended to provide students with experiences in using computer programming techniques and skills to solve problems involving mathematical models. Students enrolled in Computer Mathematics are assumed to have studied the concepts and skills in Algebra 1 and beginning Geometry.

Even though computer ideas should be introduced in the context of mathematical concepts, problem solving should be developed in the most general sense, allowing the techniques to be applied by students in many other environments. Strategies include defining the problem; developing, refining, and implementing a plan; and testing and revising the solution. Programming, ranging from simple programs involving only a few lines to complex programs involving subprograms, should permeate the entire course. Programming concepts, problem-solving strategies, and mathematical applications should be integrated throughout the course.

These standards identify fundamental principles and concepts in the field of computer science that will be used within the context of mathematical problem solving in a variety of applications. As students develop and refine skills in logic, organization, and precise expression, they will apply those skills to enhance learning in all disciplines.

Data Representation and Storage

CM.DRS.1 The student will represent data and convert data between different number systems.

Students will demonstrate the following Knowledge and Skills:

a) Represent data in different number systems, including binary, decimal, and hexadecimal.
b) Convert data between number systems (e.g., binary to decimal, decimal to hexadecimal).

CM.DRS.2 The student will differentiate between variable data types based upon their characteristics.

Students will demonstrate the following Knowledge and Skills:

a) Describe the characteristics of different variable data types, including
   i) Boolean;
   ii) character;
   iii) integer;
   iv) decimal (double/float); and
   v) string.
b) Differentiate between variable data types to determine the data type needed based upon intended use (e.g., character versus string, integer versus double/float).

CM.DRS.3 The student will represent data using appropriate data structures.

Students will demonstrate the following Knowledge and Skills:

a) Given a specific task or problem, determine the appropriate data structure (e.g., lists, arrays, objects) to represent data.

b) Perform tasks related to lists or arrays (one-dimensional or two-dimensional), including
    i) declare a list or array (one-dimensional or two-dimensional);
    ii) choose an appropriate data type for a list or an array; and
    iii) fill the list or array with data.
c) Access and manipulate a particular element of a list or an array.
d) Implement predefined objects to consolidate related information of different data types.

Components of Programming

CM.CP.1 The student will design a step-by-step plan to perform a task or solve a problem, including those arising from mathematical or interdisciplinary contexts.

Students will demonstrate the following Knowledge and Skills:

a) Design a step-by-step plan to perform a task or solve a problem using a flowchart or pseudocode that outlines the subtasks needed.
b) Define the variables needed to perform a task or solve a problem.
c) Define the constraints of a task or problem (e.g., pre-conditions, post-conditions) to determine the desired input and output.

CM.CP.2 The student will construct Boolean expressions and implement conditional statements.

Students will demonstrate the following Knowledge and Skills:

a) Write and implement Boolean expressions using logical and relational operators (e.g., !, &&, ||, ==, <, >, >=, <=, !=).
b) Write and implement "if" conditional statements.
c) Write and implement "if/else" conditional statements.
d) Write and implement compound conditional statements (e.g., nested conditionals, chained conditional statements).
e) Determine which parts of an algorithm are executed based on a condition being true or false.

CM.CP.3 The student will perform iteration with loops.

Students will demonstrate the following Knowledge and Skills:

a) Write and implement "while" and "for" loops.
b) Differentiate between loops that run a fixed number of times and loops that run an indefinite number of times (e.g., stopping dependent on variable conditions).
c) Identify conditions that cause infinite loops.
d) Determine the outcome of code segments that include loops.

CM.CP.4 The student will write and implement the output phase of a computer program.

Students will demonstrate the following Knowledge and Skills:

a) Write and implement the output phase of a computer program, which may include:
    i) formatting output in text-based environments;

ii) displaying output through a graphical user interface; and

iii) sending output to a physical device (e.g., speakers, robots, LED lights).

b) Write output to a file.

CM.CP.5 The student will write and implement the input phase of a computer program.

Students will demonstrate the following Knowledge and Skills:

a) Write and implement input statements to store user given values into a program.

b) Validate input data using exception coding (e.g., using a "while" loop to control valid input by a user).

c) Determine what output a program will produce given a specific input.

CM.CP.6 The student will implement library functions.

Students will demonstrate the following Knowledge and Skills:

a) Implement library functions to process data.

b) Implement library functions to perform mathematical operations (e.g., random, absolute value, square root, power).

c) Implement void library functions and return library functions.

d) Implement overloaded library functions.

CM.CP.7 The student will write and implement user-defined functions.

Students will demonstrate the following Knowledge and Skills:

a) Write and implement a user-defined function to complete a task or sub-task.

b) Write and implement void functions and return functions.

c) Write and implement functions that accept parameters.

CM.CP.8 The student will implement pre-defined algorithms, including search routines and sort routines.

Students will demonstrate the following Knowledge and Skills:

a) Differentiate between types of search routines.

b) Differentiate between types of sort routines.

c) Implement pre-defined algorithms.

d) Implement a search routine on a one-dimensional list or an array, including sequential search and binary search.

e) Implement a sort routine on a one-dimensional list or an array (e.g., selection sort, insertion sort, merge sort).

Applications of Programming

CM.AP.1 The student will write and implement programs using sequencing, selection, and iteration to perform a specific task or solve a problem, including those arising from mathematical and interdisciplinary contexts.

Students will demonstrate the following Knowledge and Skills:

a) Determine what components of programming are needed to implement a step-by-step plan to perform a specific task or solve a problem.
b) Write a computer program that includes sequencing, selection (conditionals), and iteration (loops).
c) Write and implement computer programs to solve mathematical problems using
    i) formulas and equations;
    ii) functions;
    iii) probability and statistics; and
    iv) data-analysis.

CM.AP.2 The student will create documentation using written comments to annotate the intended purpose of the components of a user-created program.

Students will demonstrate the following Knowledge and Skills:

a) Create documentation using written comments to:
    i) describe the overall purpose of a program;
    ii) align a previously created step-by-step plan to a written program;
    iii) describe pre-conditions and post-conditions; and
    iv) improve the readability of a program.

CM.AP.3 The student will verify how programs access and process variables.

Students will demonstrate the following Knowledge and Skills:

a) Verify that the variable types are aligned to the purpose of the algorithm.
b) Verify that global variables are set to constant values before run time.
c) Differentiate between the scopes of variables (e.g., global scope versus local scope) and verify the intended use.

CM.AP.4 The student will translate a mathematical expression or statement into computer code.

Students will demonstrate the following Knowledge and Skills:

a) Declare, initialize, and assign variables to represent mathematical expressions or statements.
b) Implement order of operations, including logical and relational operators.
c) Translate a mathematical expression or statement into a programming statement(s).

CM.AP.5 The student will trace existing code to interpret the intended purpose.

Students will demonstrate the following Knowledge and Skills:

a) Trace existing code of an algorithm to
    i) identify values at each stage of an algorithm; and
    ii) predict return values of functions given specific arguments.
b) Use tracing to describe the intended purpose of existing code for an algorithm.

Evaluation of Programming

CM.EP.1 The student will test a program to match a sample output, using a set of data.

Students will demonstrate the following Knowledge and Skills:

a) Produce a given output by entering a data set.
b) Test a program including boundary cases and inaccurate data types to verify the intended outcomes.

CM.EP.2 The student will identify errors and debug a program using various techniques.

Students will demonstrate the following Knowledge and Skills:

a) Differentiate among syntax errors, runtime errors, and logic errors.
b) Debug a program using various techniques:
    i) interpret syntax and runtime error messages;
    ii) place controlled breaks;
    iii) output intermediate results;
    iv) disable a section of code by converting it into a comment;
    v) trace code to identify logic errors; and
    vi) use debugging tools available in the programming environment.

CM.EP.3 The student will compare and contrast the efficiency of computer programs.

Students will demonstrate the following Knowledge and Skills:

a) Compare and contrast the efficiency of computer programs in terms of
    i) complexity of algorithms with the same intended outcomes;
    ii) memory space used; and
    iii) run time.