

Computer Mathematics

This course is intended to provide students with experiences in using computer programming techniques and skills to solve problems that can be set up as mathematical models. Students enrolled in Computer Mathematics are assumed to have studied the concepts and skills in Algebra I 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, making the techniques applicable 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.

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.

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.

COM.4 The student will design an algorithm to solve a given problem.

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.

COM.7 The student will select and call library functions to process data, as appropriate.

COM.8 The student will implement conditional statements that include “if/then” statements, “if/then/else” statements, case statements, and Boolean logic.

COM.9 The student will implement pre-defined algorithms, including sort routines, search routines, and simple animation routines.

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.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.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.
- COM.14 The student will select and implement appropriate data structures, including arrays (one- and/or two-dimensional), and objects.

Data Manipulation

- 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.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.

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.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.