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. Students who successfully complete the standards for this course may earn credit toward meeting the mathematics graduation requirement. It is recognized that many students will gain computer skills in other mathematics courses or in a separate curriculum outside of mathematics and prior to high school. In such cases, the standards indicated by an asterisk (*) should be included in the student's course of study and treated as a review.

Even though computer ideas should be introduced in the context of mathematical concepts, problem solving per se 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 and may include programming a graphing calculator or scripting a problem solution in a database or spreadsheet. 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.

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.

*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.

*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.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.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.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).

*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.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.9 The student will define simple variable data types that include integer, real (fixed and scientific notation), character, string, and Boolean.

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.

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.12 The student will translate a mathematical expression into a computer statement, which involves writing assignment statements and using the order of operations.

COM.13 The student will select and implement built-in (library) functions in processing data.

COM.14 The student will implement conditional statements that include "if/then" statements, "if/then/else" statements, case statements, and Boolean logic.

COM.15 The student will implement loops, including iterative loops. Other topics will include single entry point, single exit point, preconditions, and postconditions.

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.

*COM.17 The student will implement pre-existing algorithms, including sort routines, search routines, and simple animation routines.

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.

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.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.