

**Computer Mathematics**

**Mathematics  
Curriculum Framework**

**Revised 2004**

Course Title: Computer Mathematics (Fourth-Year Course)  
Course/Unit Credit: 1  
Course Number:  
Teacher Licensure: Secondary Mathematics  
Pre-requisite: Currently Enrolled or Pre-requisite Algebra II  
Grades: 9-12

### Computer Mathematics

Computer Mathematics provides students with experiences in using the computer to solve problems that can be set up as mathematical models. Students entering this course must be currently enrolled in or have already taken Algebra II. Students will develop and refine skills in logic, organization, and precise expression, thereby enhancing learning in other disciplines. Programming should be introduced in the context of mathematical concepts and problem solving. Students will define a problem; develop, refine, and implement a plan; and test and revise the solution. It is recommended that class size be no larger than 20 students because of the hands-on projects and activity nature of the course.

Teachers should help students make connections to other subject areas and fields of endeavor through applications. Using manipulatives, graphing calculators, and computer spread sheet applications should help students develop and attach meaning to abstract ideas.

Strand	Standard
Problem Solving	
	1. The student will develop and apply logical reasoning skills to solve real-world problems through the development of mathematical models.
Program Design	
	2. The student will design a step-by-step plan to solve a given problem.
Program Implementation	
	3. The student will use worksheet functions to program using a computer spreadsheet application program.
	4. The student will use the programming tool to create programs using a programmable calculator.
Data Manipulation and Testing	
	5. The student will manipulate data to adjust and test programs designed in a computer spreadsheet application.
	6. The student will manipulate data to adjust and test programs designed in a programmable calculator.

## Problem Solving

Content Standard 1. The student will develop and apply logical reasoning skills to solve real-world problems through the development of mathematical models.

PS.1.CM.1	Analyze and interpret graphs, charts, and tables in the design and implementation of a computer program.
PS.1.CM.2	Write an <i>algorithm</i> to solve mathematical problems using formulas, equations, and functions.
PS.1.CM.3	Analyze and interpret truth tables from basic statements using <i>Boolean operators</i> (AND, OR, XOR, and NOT).
PS.1.CM.4	Write an <i>algorithm</i> from a mathematical model.

## Program Design

Content Standard 2. The student will design a step-by-step plan to solve a given problem.

PD.2.CM.1	Translate a mathematical expression into a computer statement, which involves writing assignment statements and using the order of operations.
PD.2.CM.2	Implement conditional statements that include if/then, if/then/else, case statements, and <i>Boolean logic</i> .
PD.2.CM.3	Define and differentiate Decision (selection) and Sequence (process).
PD.2.CM.4	The student will represent an algorithm representation as a <i>flowchart</i> and in <i>pseudocode</i> .
PD.2.CM.5	Use <i>flowchart</i> terminology, such as terminals (starts and stops), <i>subroutines</i> , and connectors.
PD.2.CM.6	Develop <i>recursive</i> relationships from mathematical models (e.g. arithmetic and geometric sequences).
PD.2.CM.7	Define and use <i>variable data types</i> (integers, real, character).

## Program Implementation

Content Standard 3. The student will use worksheet functions to program using a computer spreadsheet application program.

PI.3.CM.1	Using a spreadsheet program, create an <i>array</i> .
PI.3.CM.2	Create functions using <i>recursions</i> and <i>loops</i> .
PI.3.CM.3	Locate, categorize, and implement worksheet functions.
PI.3.CM.4	Create <i>constraints</i> to validate <i>cell</i> entries.
PI.3.CM.5	Using a spreadsheet program, <i>sort</i> data using various methods (e.g. <i>bubble</i> , <i>quick</i> , and <i>shell</i> ).

## Program Implementation

Content Standard 4. The student will use the programming tool to create programs using a programmable calculator.

PI.4.CM.1	Create, edit, and execute a program utilizing an <i>array</i> .
PI.4.CM.2	Create, edit, and execute programs using <i>loops</i> .
PI.4.CM.3	Create, edit, and execute programs to calculate mathematical formulas, such as the quadratic formula, and volume of a simple solid.
PI.4.CM.4	Develop functional programs from algorithms developed from the mathematical models.
PI.4.CM.5	Create programs using various display modes (including tables and graphs).
PI.4.CM.6	Locate, categorize, and implement programming commands.
PI.4.CM.7	Use <i>subroutines</i> to reduce keystrokes and memory use.

## Data Manipulation and Testing

Content Standard 5. The student will manipulate data to adjust and test programs designed in a computer spreadsheet application.

DMT.5.CM.1	Name a <i>range</i> (one cell or a group of cells), and use the name to select cells.
DMT.5.CM.2	Using the Scenario tool, estimate best-case or worst-case scenarios.

## Data Manipulation and Testing

Content Standard 6. The student will manipulate data to adjust and test programs designed in a programmable calculator.

DMT.6.CM.1	Compare results from mathematical formulas to their program equivalent.
DMT.6.CM.2	Identify and eliminate error messages using troubleshooting techniques ( <i>debug</i> ).
DMT.6.CM.3	Understand and differentiate the different error types ( <i>syntax</i> , <i>runtime</i> , and <i>logic</i> ).
DMT.6.CM.4	Design and investigate best-case or worst-case scenarios of a program.

## Computer Math Glossary

<b>Algorithm</b>	A formula or set of steps for solving a particular problem
<b>Array</b>	A collection of data items, all of the same type, in which each item's position is uniquely designed by an integer.
<b>Boolean logic</b>	Boolean logic is a form of algebra in which all values reduced to either TRUE or FALSE (Boolean logic is especially important for computer science because it fits nicely with the binary numbering system, in which each bit has a value of either 1 or 0.)
<b>Bubble sort</b>	Sort by comparing each adjacent pair of items in a list, swapping the items if necessary, and repeating the pass through until no swaps are done
<b>Cell</b>	In spreadsheets, the intersection of a row and a column
<b>Constraints</b>	A condition or proposition that must be maintained as true
<b>Data types</b>	Specifies and limits the kind of data that may be entered into a field
<b>Debug</b>	Find and remove programming errors (runtime, syntax, and logic)
<b>Filter</b>	A program or routine that accepts certain data as input, transforms it, and then outputs the transformed data
<b>Flowchart</b>	A graphic structured representation of the major steps in a process
<b>Logic</b>	A mathematical treatment of formal logic whereby a system of symbols (AND, OR, and NOT) is used to represent quantities and relationships
<b>Logical operators</b>	Perform a comparison between two values and produce a result of true or false, or 1 or 0
<b>Loops</b>	A single execution of a set of instructions that are repeated until a certain condition is met
<b>Nested</b>	A set of predefined instructions contained within a routine or program
<b>Process</b>	An instance of a running program
<b>Program</b>	An organized list of instructions that, when executed, causes the computer to behave in a predetermined manner
<b>Pseudocode</b>	An English language version of an algorithm that will ultimately be translated into real computer code
<b>Quick sort</b>	Selection of an element (which becomes a pivot) from the array, partitions the remaining elements into greater and less than the pivot, and recursively sorts the partitions
<b>Range</b>	The limits of the values a function can take
<b>Recursive</b>	A program or routine that calls itself continuously until a certain condition is met
<b>Runtime</b>	The period of time during which a program is executing
<b>Shell sort</b>	The shell sorting algorithm makes multiple passes through the list, and each time sorts a number of equally sized sets
<b>Sort</b>	Arrange items in a predetermined order
<b>Subroutine</b>	A short program segment that performs a specific function and is available for general use by other programs and routines
<b>Syntax</b>	The rules for exactly how symbols and words can be combined within a particular programming language

