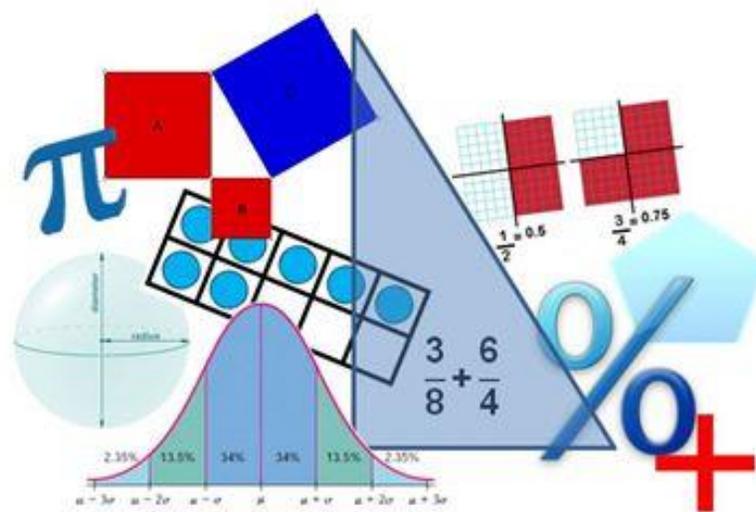


# Mathematics

## 2016 Standards of Learning

### Computer Mathematics Curriculum Framework



Board of Education  
Commonwealth of Virginia

Copyright © 2016  
by the  
Virginia Department of Education  
P.O. Box 2120  
Richmond, Virginia 23218-2120  
<http://www.doe.virginia.gov>

All rights reserved. Reproduction of these materials for instructional purposes in public school classrooms in Virginia is permitted.

**Superintendent of Public Instruction**  
Steven R. Staples

**Chief Academic Officer/Assistant Superintendent for Instruction**  
Steven M. Constantino

**Office of Mathematics and Governor's Schools**  
Debra Delozier, Mathematics Specialist  
Tina Mazzacane, Mathematics and Science Specialist  
Christa Southall, Mathematics Specialist

**Acknowledgements**

The Virginia Department of Education wishes to express sincere thanks to Michael Bolling, who assisted in the development of the 2016 *Mathematics Standards of Learning* and 2016 *Mathematics Standards of Learning Curriculum Framework*.

**NOTICE**

The Virginia Department of Education does not unlawfully discriminate on the basis of race, color, sex, national origin, age, or disability in employment or in its educational programs or services.

# Virginia 2016 *Mathematics Standards of Learning Curriculum Framework*

## Introduction

The 2016 *Mathematics Standards of Learning Curriculum Framework*, a companion document to the 2016 *Mathematics Standards of Learning*, amplifies the *Mathematics Standards of Learning* and further defines the content knowledge, skills, and understandings that are measured by the Standards of Learning assessments. The standards and *Curriculum Framework* are not intended to encompass the entire curriculum for a given grade level or course. School divisions are encouraged to incorporate the standards and *Curriculum Framework* into a broader, locally designed curriculum. The *Curriculum Framework* delineates in greater specificity the minimum content that all teachers should teach and all students should learn. Teachers are encouraged to go beyond the standards as well as to select instructional strategies and assessment methods appropriate for all students.

The *Curriculum Framework* also serves as a guide for Standards of Learning assessment development. Students are expected to continue to connect and apply knowledge and skills from Standards of Learning presented in previous grades as they deepen their mathematical understanding. Assessment items may not and should not be a verbatim reflection of the information presented in the Curriculum Framework.

Each topic in the 2016 *Mathematics Standards of Learning Curriculum Framework* is developed around the Standards of Learning. The format of the *Curriculum Framework* facilitates teacher planning by identifying the key concepts, knowledge, and skills that should be the focus of instruction for each standard. The *Curriculum Framework* is divided into two columns: Understanding the Standard and Essential Knowledge and Skills. The purpose of each column is explained below.

### *Understanding the Standard*

This section includes mathematical content and key concepts that assist teachers in planning standards-focused instruction. The statements may provide definitions, explanations, examples, and information regarding connections within and between grade level(s)/course(s).

### *Essential Knowledge and Skills*

This section provides a detailed expansion of the mathematics knowledge and skills that each student should know and be able to demonstrate. This is not meant to be an exhaustive list of student expectations.

## **Mathematical Process Goals for Students**

---

The content of the mathematics standards is intended to support the following five process goals for students: becoming mathematical problem solvers, communicating mathematically, reasoning mathematically, making mathematical connections, and using mathematical representations to model and interpret practical situations. Practical situations include real-world problems and problems that model real-world situations.

### **Mathematical Problem Solving**

Students will apply mathematical concepts and skills and the relationships among them to solve problem situations of varying complexities. Students also will recognize and create problems from real-world data and situations within and outside mathematics and then apply appropriate strategies to determine acceptable solutions. To accomplish this goal, students will need to develop a repertoire of skills and strategies for solving a variety of problems. A major goal of the mathematics program is to help students apply mathematics concepts and skills to become mathematical problem solvers.

### **Mathematical Communication**

Students will communicate thinking and reasoning using the language of mathematics, including specialized vocabulary and symbolic notation, to express mathematical ideas with precision. Representing, discussing, justifying, conjecturing, reading, writing, presenting, and listening to mathematics will help students clarify their thinking and deepen their understanding of the mathematics being studied. Mathematical communication becomes visible where learning involves participation in mathematical discussions.

### **Mathematical Reasoning**

Students will recognize reasoning and proof as fundamental aspects of mathematics. Students will learn and apply inductive and deductive reasoning skills to make, test, and evaluate mathematical statements and to justify steps in mathematical procedures. Students will use logical reasoning to analyze an argument and to determine whether conclusions are valid. In addition, students will use number sense to apply proportional and spatial reasoning and to reason from a variety of representations.

### **Mathematical Connections**

Students will build upon prior knowledge to relate concepts and procedures from different topics within mathematics and see mathematics as an integrated field of study. Through the practical application of content and process skills, students will make connections among different areas of mathematics and between mathematics and other disciplines, and to real-world contexts. Science and mathematics teachers and curriculum writers are encouraged to develop mathematics and science curricula that support, apply, and reinforce each other.

### **Mathematical Representations**

Students will represent and describe mathematical ideas, generalizations, and relationships using a variety of methods. Students will understand that representations of mathematical ideas are an essential part of learning, doing, and communicating mathematics. Students should make connections among different representations – physical, visual, symbolic, verbal, and contextual – and recognize that representation is both a process and a product.

## **Instructional Technology**

---

The use of appropriate technology and the interpretation of the results from applying technology tools must be an integral part of teaching, learning, and assessment. However, facility in the use of technology shall not be regarded as a substitute for a student’s understanding of quantitative and algebraic concepts and relationships or for proficiency in basic computations. Students must learn to use a variety of methods and tools to compute, including paper and pencil, mental arithmetic, estimation, and calculators. In addition, graphing utilities, spreadsheets, calculators, dynamic applications, and other technological tools are now standard for mathematical problem solving and application in science, engineering, business and industry, government, and practical affairs.

Calculators and graphing utilities should be used by students for exploring and visualizing number patterns and mathematical relationships, facilitating reasoning and problem solving, and verifying solutions. However, according to the National Council of Teachers of Mathematics, “... the use of calculators does not supplant the need for students to develop proficiency with efficient, accurate methods of mental and pencil-and-paper calculation and in making reasonable estimations.” State and local assessments may restrict the use of calculators in measuring specific student objectives that focus on number sense and computation. On the grade three state assessment, all objectives are assessed without the use of a calculator. On the state assessments for grades four through seven, objectives that are assessed without the use of a calculator are indicated with an asterisk (\*).

## **Computational Fluency**

---

Mathematics instruction must develop students’ conceptual understanding, computational fluency, and problem-solving skills. The development of related conceptual understanding and computational skills should be balanced and intertwined, each supporting the other and reinforcing learning.

Computational fluency refers to having flexible, efficient and accurate methods for computing. Students exhibit computational fluency when they demonstrate strategic thinking and flexibility in the computational methods they choose, understand and can explain, and produce accurate answers efficiently.

The computational methods used by a student should be based on the mathematical ideas that the student understands, including the structure of the base-ten number system, number relationships, meaning of operations, and properties. Computational fluency with whole numbers is a goal of mathematics instruction in the elementary grades. Students should be fluent with the basic number combinations for addition and subtraction to 20 by the end of second grade and those for multiplication and division by the end of grade four. Students should be encouraged to use computational methods and tools that are appropriate for the context and purpose.

## **Algebra Readiness**

---

The successful mastery of Algebra I is widely considered to be the gatekeeper to success in the study of upper-level mathematics. “Algebra readiness” describes the mastery of, and the ability to apply, the *Mathematics Standards of Learning*, including the Mathematical Process Goals for Students, for kindergarten through grade eight. The study of algebraic thinking begins in kindergarten and is progressively formalized prior to the study of the algebraic content found in the Algebra I Standards of Learning. Included in the progression of algebraic content is patterning, generalization of arithmetic concepts, proportional reasoning, and representing mathematical relationships using tables, symbols, and graphs. The K-8 *Mathematics Standards of Learning* form a progression of content knowledge and develop the reasoning necessary to be well-prepared for mathematics courses beyond Algebra I, including Geometry and Statistics.

## Equity

---

“Addressing equity and access includes both ensuring that all students attain mathematics proficiency and increasing the numbers of students from all racial, ethnic, linguistic, gender, and socioeconomic groups who attain the highest levels of mathematics achievement.”

– National Council of Teachers of Mathematics

Mathematics programs should have an expectation of equity by providing all students access to quality mathematics instruction and offerings that are responsive to and respectful of students’ prior experiences, talents, interests, and cultural perspectives. Successful mathematics programs challenge students to maximize their academic potential and provide consistent monitoring, support, and encouragement to ensure success for all. Individual students should be encouraged to choose mathematical programs of study that challenge, enhance, and extend their mathematical knowledge and future opportunities.

Student engagement is an essential component of equity in mathematics teaching and learning. Mathematics instructional strategies that require students to think critically, to reason, to develop problem-solving strategies, to communicate mathematically, and to use multiple representations engages students both mentally and physically. Student engagement increases with mathematical tasks that employ the use of relevant, applied contexts and provide an appropriate level of cognitive challenge. All students, including students with disabilities, gifted learners, and English language learners deserve high-quality mathematics instruction that addresses individual learning needs, maximizing the opportunity to learn.

**COM.1 The student will design and apply computer programs to solve practical problems in mathematics arising from business and applications in mathematics.**

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• Programming languages require the use of particular structures to express algorithms as programs.</li> <li>• Designing algorithms is a problem solving phase of computer programming.</li> <li>• Practical problems that can be modeled mathematically can be solved with a computer program.</li> <li>• Sample mathematical problems may include:                             <ul style="list-style-type: none"> <li>– the bisection method for a solution to a polynomial equation;</li> <li>– use of recursion to solve the Tower of Hanoi problem;</li> <li>– implementing a cellular automaton such as Conway’s Game of Life;</li> <li>– implementing a maze-traversing or map-coloring algorithm; and</li> <li>– determining whether an input number is prime.</li> </ul> </li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Design and implement computer programs to solve practical problems.</li> <li>• Analyze and interpret graphs, charts, and tables in the design and implementation of a computer program.</li> <li>• Design and implement computer programs to                             <ul style="list-style-type: none"> <li>– solve practical problems arising from business; and</li> <li>– solve mathematical problems, using formulas, equations, and functions.</li> </ul> </li> </ul>

**COM.2 The student will design, write, document, test, and debug a computer program.**

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• Programming documentation includes the step-by-step plan, the test data, a sample run, and the program listing with appropriately placed comments.</li> <li>• Documentation enables programmers to use modules written by others without having to read and understand their code.</li> <li>• Comments communicate a programmer's original intent and are useful for debugging incorrect code.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Describe a computer program.</li> <li>• Design, write, document, test, and debug a complete computer program.</li> <li>• List and describe the processes involved in writing a computer program.</li> <li>• Describe the function of an algorithm.</li> <li>• Provide required documentation for a program.</li> <li>• Determine what a given output statement will print.</li> <li>• Debug a program.</li> </ul>

**COM.3** The student will write program specifications that define the constraints of a given problem.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• A programmer begins the programming process by analyzing the problem and developing a general solution (algorithm).</li> <li>• The successful completion of a computer program requires problem solving skills.</li> <li>• Program specifications include descriptions of preconditions, postconditions, the desired input and output, analysis of the available input, and an indication as to whether or not the problem is solvable using a computer program.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Write program specifications that define the constraints of a given problem.</li> <li>• Describe the preconditions, postconditions, and desired input and output of a given problem.</li> <li>• Determine whether or not a given problem is solvable using a computer program.</li> </ul>

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

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"><li>• All programs are implementations of algorithms.</li><li>• Algorithms used to solve problems may include flowcharts, pseudo code, and/or data-flow diagrams.</li></ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"><li>• Design a step-by-step algorithm to solve a problem.</li></ul>

**COM.5** The student will divide a given problem into modules by task and implement the solution.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• Functional decomposition is a way to develop a program in which the problem is divided into sub-problems whose solutions comprise the solution to the original problem.</li> <li>• A subtask that has been solved previously may be reused in other programs.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Divide a problem into modules by task.</li> <li>• Write task-oriented modules, which may include               <ul style="list-style-type: none"> <li>– a user-defined function;</li> <li>– subroutines; or</li> <li>– procedures.</li> </ul> </li> <li>• Determine the need for a subroutine or user-defined function.</li> <li>• Determine the difference between and the need for internal and external subroutines and functions.</li> <li>• Implement the solution of the problem.</li> </ul>

**COM.6** The student will translate mathematical expressions into programming expressions by declaring variables, writing assignment statements, and using the order of operations.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"><li>• A variable identifies a location in memory where a data value that can be changed is stored.</li><li>• An assignment statement stores the value of an expression into a variable.</li></ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"><li>• Translate mathematical expressions into a programming expression.</li><li>• Declare appropriately named variables to store values used in computations.</li><li>• Write variable assignment statements.</li><li>• Use the order of operations to simplify expressions.</li><li>• Construct and evaluate expressions that include multiple arithmetic operations.</li></ul>

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

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"><li>• The argument of a library function is a value or expression associated with the independent variable.</li><li>• A library function is a subroutine.</li></ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"><li>• Use library functions in designing programs to process data.</li><li>• Use library functions that are arithmetic or string operations.</li><li>• Invoke a value-returning library function.</li></ul>

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

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• Boolean logic is a system using variables with only two values: TRUE and FALSE.</li> <li>• The “if” statement is the fundamental control structure that allows branches in the flow of control.</li> <li>• Case statements may be used to simplify a series of “if/then/else” statements.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Construct a simple logical (Boolean) expression to evaluate a given condition.</li> <li>• Construct an “if/then” statement to perform a specific task.</li> <li>• Construct an “if/then/else” statement to perform a specific task.</li> <li>• Construct a case statement to perform a specific task.</li> <li>• Use conditional statements to incorporate decision making into programs.</li> </ul>

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

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"><li>• Arranging values into an order is known as sorting.</li><li>• A search algorithm finds an item with specified properties among a collection of items.</li></ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"><li>• Implement pre-defined algorithms into a program.</li><li>• Implement a sort routine on a one-dimensional array.</li><li>• Implement a sequential search routine on a one-dimensional array.</li><li>• Implement a binary search routine on a one-dimensional array.</li><li>• Implement a simple animation routine.</li></ul>

**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.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• A program needs data on which to operate.</li> <li>• A file is a named area in secondary storage that holds a collection of information.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Design a screen layout to facilitate input.</li> <li>• Design program information input by user interaction and file input.</li> <li>• Validate data, using a variety of methods.</li> <li>• Construct input statements to read values into a program.</li> <li>• Determine the contents of variables that have been assigned values by input statements.</li> </ul>

**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.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"><li>• Output is dependent on input and processing.</li><li>• Implementation of the output portion of a program includes designing the output layout and displaying the output in the desired format.</li></ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"><li>• Design an output layout.</li><li>• Access available output devices.</li><li>• Use output statements.</li><li>• Label results.</li></ul>

**COM.12** The student will design and implement computer graphics to enhance output.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"><li>• Work with computer graphics is specific to the programming language.</li></ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"><li>• Design and implement computer graphics using various techniques such as<ul style="list-style-type: none"><li>– plotting points or shapes;</li><li>– determining and setting window or screen dimensions;</li><li>– determining and setting screen or background colors;</li><li>– using box commands.</li></ul></li><li>• Describe the role of graphics in the computer environment.</li></ul>

**COM.13** The student will implement various mechanisms for performing iteration with an algorithm.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• An iterative algorithm executes a sequence of statements repeatedly.</li> <li>• Iterative algorithms execute a fixed number of times.</li> <li>• Iterative algorithms can be nested.</li> <li>• Iterative algorithms include               <ul style="list-style-type: none"> <li>– iterative loops;</li> <li>– indefinite loops;</li> <li>– pretest loops;</li> <li>– posttest loops; and</li> <li>– nested loops.</li> </ul> </li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Determine when an iterative algorithm is needed in a computer program.</li> <li>• Incorporate single entry point, single exit point, preconditions, and postconditions into iterative algorithms.</li> </ul>

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

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• Data structures are collections of components that are given a single name and whose organization is characterized by the method used to access the individual components.</li> <li>• Two-dimensional arrays may be viewed as arrays of one-dimensional arrays.</li> <li>• Data files may be used to store information needed each time the program is executed.</li> <li>• An object is a basic data structure used to organize data of different types and referenced by name.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Implement a one-dimensional or two-dimensional array for a given problem.               <ul style="list-style-type: none"> <li>– Choose an appropriate data type for an array.</li> <li>– Assign a value to an array element.</li> <li>– Fill an array with data, and process the data in the array.</li> <li>– Access a particular element of a two-dimensional array.</li> <li>– Process a two-dimensional array by rows and by columns.</li> <li>– Retrieve data from an array.</li> </ul> </li> <li>• Use data files in computer programs, both as a source of input data and as a way to save data for the next program execution.</li> <li>• Implement objects to consolidate related information of different data types.</li> </ul>

**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.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• A data type defines a set of values and a set of operations on the values.</li> <li>• Boolean data has only two literal constants, and they represent TRUE and FALSE.</li> <li>• A string is an array of character data for which there exists an aggregate name.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Define variables using data types, including               <ul style="list-style-type: none"> <li>– integer;</li> <li>– real (fixed and scientific notation);</li> <li>– character;</li> <li>– string;</li> <li>– Boolean; and</li> <li>– object.</li> </ul> </li> <li>• Use standard naming conventions to create variable names.</li> </ul>

**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.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>• Computers consist of hardware components that interact with software.</li> <li>• Program design must identify each piece of data as a constant, local variable, or global variable depending on the scope of the data in the program.</li> <li>• Parameters are used to provide information to other parts of the program (functions, methods, subroutines, or procedures).</li> <li>• Parameter passing may include using variables' addresses or pointers or equivalent features.</li> <li>• Variable addresses and pointers are used to access data stored in computer memory.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>• Determine when the use of a variable is appropriate.</li> <li>• Describe how a computer stores, accesses, and processes variables.</li> <li>• Incorporate parameter passing into programs.</li> <li>• Differentiate between local and global variables and describe their appropriate use.</li> <li>• Compare and contrast variables and constants.</li> <li>• Describe the basic interplay between hardware and software in program execution.</li> </ul>

**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.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"><li>• A program test can reveal problems (bugs) in the program.</li><li>• Testing a program for bugs is part of problem solving.</li><li>• Various forms of data can be used to debug a program.</li></ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"><li>• Test a program, using an appropriate and complete set of data.</li><li>• Demonstrate that a set of data tests all branches of a program.</li></ul>

**COM.18** The student will debug a program using appropriate techniques (e.g., appropriately placed controlled breaks, the printing of intermediate results, and other debugging tools available in the programming environment), and identify the difference among syntax errors, runtime errors, and logic errors.

Understanding the Standard	Essential Knowledge and Skills
<ul style="list-style-type: none"> <li>Debugging is a methodical process of finding and correcting errors so the computer program executes as intended.</li> </ul>	<p><b>The student will use problem solving, mathematical communication, mathematical reasoning, connections, and representations to</b></p> <ul style="list-style-type: none"> <li>Debug a program, using controlled breaks, the printing of intermediate results, and other debugging tools.</li> <li>Identify the differences among syntax errors, runtime errors, and logic errors.</li> </ul>