High School: Computer Science and Mathematics

Overview

The standards address the skills, knowledge, and dispositions that students should develop to foster mathematical understanding and expertise, as well as concepts, skills, and knowledge – what students need to understand, know, and be able to do. The standards also require that the Mathematical Habits of Mind and course-specific Mathematics Content Standards be connected. These connections are essential to support the development of students' broader mathematical understanding, as students who lack understanding of a topic may rely too heavily on procedures. The Mathematical Habits of Mind must be taught as carefully and practiced as intentionally as the course-specific Mathematics Content Standards are. Neither type should be isolated from the other; mathematics instruction is most effective when these two aspects of the West Virginia College- and Career-Readiness Standards for Mathematics come together as a powerful whole.

Mathematical Habits of Mind

WEST VIRGINIA COLLEGE- AND CAREER-READINESS STANDARDS FOR TECHNOLOGY AND COMPUTER SCIENCE (2520.14)

High School: Computer Science & Mathematics

This introduction to programming course is designed to provide students with the opportunity to explore the uses of mathematics and computer programming as tools in creating effective solutions to complex problems. Students will develop and refine fundamental skills of computer science within a mathematical context.

Computer Science & Mathematics may be counted as a fourth math elective credit course and must be taught by a certified 9-12 math teacher. Any reference to an algorithm or algorithms in this document includes both mathematics and computer science contexts. Throughout the course, students will use developmentally appropriate and accurate terminology when communicating about technology. Teachers are responsible for including the eight Standards for Mathematical Practice. The teacher or local district may select the object-oriented programming language(s) used in the course.

Prerequisite:
Successful completion of High School Math II or High School Geometry before starting this course. No previous computer science course is required.

Computer Systems and Computational Thinking	
Evaluate different data representations to solve problems.	
CS.M.1	Analyze the various mathematical bases (e.g., binary, decimal, hexadecimal) and convert between them.
CS.M.2	Describe the relationship between binary and hexadecimal representations.
CS.M.3	Convert information between various encoding formats (e.g., ASCII, Unicode, hexadecimal, binary).
CS.M.4	Compare techniques (e.g., sorting, statistics, searching) for analyzing massive data collections.
Connect the development cycle of algorithm construction to problem-solving.	
CS.M.5	Describe how mathematical and statistical functions, sets, and logic are used in computation.
CS.M.6	Utilize predefined mathematical functions and parameters to divide a complex problem into simpler parts, including parallel processing.
CS.M.7	Interpret truth tables from basic statements using Boolean operators (AND, OR, XOR, and NOT).
CS.M.8	Explain ways in which sequence, selection, iteration, and recursion are building blocks of algorithms.
CS.M.9	Create systems of equations based on real-world situations.
CS.M.10	Analyze decisions and strategies using probability and statistical concepts.
Create and evaluate algorithms to solve problems.	
CS.M.11	Utilize modeling and simulation techniques to represent and understand natural phenomena.
CS.M.12	Examine classical algorithms (e.g., searching, sorting, and shortest path).
CS.M.13	Manipulate formulas and equations and apply them to algorithm development.
CS.M.14	Apply algorithm analysis and design techniques to solve problems.
CS.M.15	Write algorithms to solve mathematical problems using formulas, equations, and functions.
CS.M.16	Implement conditional statements that include if/then, if/then/else, case statements, and Boolean logic, in the design of algorithms.
CS.M.17	Represent algorithms using flowcharts and pseudocode.
CS.M.18	Combine standard function types using arithmetic operations.
CS.M.19	Analyze algorithms for correctness, clarity, and efficiency.

Programming and Algorithms	
Evaluate the use of programming languages to solve problems and develop systems.	
CS.M.20	Compare and contrast computer programming languages and paradigms (e.g., compiled and interpreted languages, procedural and object-oriented paradigms).
CS.M.21	Diagram the program execution process.
CS.M.22	Determine the output of a given sample program without the use of a computer.
Create, test, and use computer programs to solve problems.	
CS.M.23	Implement computing applications using the following software development tools and techniques › branching (if, if-else) › declare, define, and reference variables › lists/arrays › looping (for, while, do/while) › recursion › sequencing
CS.M.24	Use various debugging and testing methods to ensure program correctness.
CS.M.25	Cite evidence to support or refute the correctness of software solutions.

Computers and Communication Devices	
Classify electronic devices containing computational processors that execute programs.	
CS.M.26	Recognize that computers are devices that execute programs.
CS.M.27	Identify a variety of electronic devices (e.g., cell phones, desktops, laptops, vehicles, programmable thermostats, and programmable kitchen appliances) that contain computational processors.
CS.M.28	Describe unique features of computers embedded in mobile devices and vehicles.
CS.M.29	Investigate the history of computers, identifying contributors and major milestones (e.g., Alan Turing, Charles Babbage, Ada Lovelace, Grace Hopper, analytical machine, ENIAC, IBM PC).
Analyze the relationship between hardware and software.	
CS.M.30	Demonstrate an understanding of the relationship between hardware and software.
CS.M.31	Develop criteria for purchasing or upgrading computer system hardware.
CS.M.32	Describe primary components of computer systems (e.g., input, output, processing, and storage).
CS.M.33	Explain multiple levels of hardware and software that support program execution (e.g., compilers, interpreters, operating systems, networks).
CS.M.34	Apply strategies for identifying and solving routine hardware problems that occur during everyday computer use.
Describe the major components and functions of networks.	
CS.M.35	Describe how the Internet facilitates global communication.
CS.M.36	Describe issues that impact network functionality (e.g., latency, bandwidth, firewalls, server capability).

Impacts of Computing	
Evaluate appropriate and inappropriate uses of technology.	
CS.M.37	Summarize appropriate and inappropriate technological behaviors, including issues of privacy, copyright, security, legalities, and politics.
CS.M.38	Explore the ramifications of inappropriate uses of technology.
CS.M.39	Investigate the national and global economic impact of cybercrime.
Investigate social and ethical issues relating to digital information.	
CS.M.40	Discuss accessibility issues (e.g., adaptive technology for special needs individuals, censorship, geographical locations, and economically-disadvantaged populations).
CS.M.41	Compare the reliability of various online sources.
CS.M.42	Investigate information ownership topics › access › distribution rights › hacking › licensure › open source › public domain › software piracy
CS.M.43	Describe security and privacy issues that relate to computer networks.
Will explore security and privacy techniques.	
CS.M.44	Explain principles of network security and techniques that protect stored and transmitted data (e.g., encryption, cryptography, and authentication).