2017 Computer Science Standards of Learning

Computer Science Programming

The Computer Science Programming standards outline the content for a one-year course with an emphasis on computer programming in a text-based language. The standards build on the concepts outlined in the Computer Science Foundations and Computer Science Principles standards.

This course continues the study of computer programming and prepares students to write programs of increasing complexity to solve problems of personal interest and professional relevance in a variety of technical fields. Additionally, this course provides the knowledge and experience to prepare students for further studies in computer science.

Teachers are encouraged to select text-based programming languages and environments, problems, challenges, and activities that are appropriate for their students to successfully meet the objectives of the standards. The majority of this course will address Algorithms and Programming. While the standards below do not include new content related to Computing Systems or Networks and the Internet, they may be used to provide context for additional exploration of these topics.

Cybersecurity

PRG.1 The student will describe and use best practices of program development that make some common flaws less likely and explain how this improves computer security.

Data and Analysis

PRG.2 The student will create programs that model the relationships among different elements in collections of real-world data.

PRG.3 The student will translate numbers between machine representations and human-accessible representations.

Algorithms and Programming

PRG.4 The student will design and implement a program working individually and in teams using a text-based language.

PRg.5 The student will explain the software life cycle and how it applies to iterative development processes.

PRG.6 The student will design and implement an algorithm
a) with compound conditional execution, and analyze and evaluate complex Boolean conditions; and

b) using complex iteration, including nested loops.

PRG.7 The student will implement programs that accept input from a variety of sources and produce output based on that input.

PRG.8 The student will trace the execution of iterative and recursive algorithms, illustrating output and changes in values of named variables.

PRG.9 The student will perform complex computations
a) on numbers, including modular division and random number generation; and
b) on strings, including substring manipulation and processing individual characters.

PRG.10 The student will demonstrate an understanding of different data types by using appropriate constructs to convert between them when appropriate.

PRG.11 The student will analyze a large-scale computational problem, identify generalizable patterns, and implement a solution.

PRG.12 The student will implement an algorithm that uses existing functions and accesses existing libraries or APIs to satisfy its requirements.

PRG.13 The student will write functions, both with and without parameters, and both with and without return values, that represent abstractions useful to the solution of a larger problem.

PRG.14 The student will create programs demonstrating an understanding of the interactions between classes in object-oriented design, and by implementing classes with instance data and methods to satisfy a design specification.

PRG.15 The student will use code written by others by reading the documentation and incorporating it into their programs using proper citation of the reused code.

PRG.16 The student will read and store data in 1D and 2D collections, and design and implement algorithms to process and manipulate those collections.

PRG.17 The student will adapt classic algorithms for use in a particular context and analyze them for effectiveness and efficiency.

PRG.18 The student will develop and use a series of test cases to verify that a program performs according to its design specifications, including edge cases and all branches.

PRG.19 The student will, through the process of code review, evaluate a program's correctness, readability, usability, and other factors.

PRG.20 The student will use a systematic approach and debugging tools to independently debug a program.

Impacts of Computing

PRG.21 The student will identify some of the practical, business, and ethical impacts of open source and free software and the widespread access they provide.