

Content Standard 1.0: Understand Algorithms and Programming

	Unit 1	Unit 2	Unit 3	Unit 4
PERFORMANCE STANDARD 1.1 : APPLY ALGORITHMS				
1.1.1 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests				
1.1.2 Describe how artificial intelligence drives many software and physical systems				
1.1.3 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem				
1.1.4 Use and adapt classic algorithms to solve computational problems				
1.1.5 Develop classic algorithms in code to solve computational problems				
1.1.6 Evaluate algorithms in terms of their efficiency, correctness, and clarity				
PERFORMANCE STANDARD 1.2 : IMPLEMENT CONTROLS				
1.2.1 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made				
1.2.2 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions				
1.2.3 Illustrate the flow of execution of a recursive algorithm				
1.2.4 Implement conditional controls in code				
1.2.5 Implement recursive algorithms in code				
PERFORMANCE STANDARD 1.3 : UTILIZE VARIABLES				
1.3.1 Demonstrate the use of both linked lists and arrays to simplify solutions, generalizing computational problems instead of repeatedly using simple variables				
1.3.2 Compare and contrast fundamental data structures and their uses				
1.3.3 Implement arrays in code				
1.3.4 Implement ArrayLists and LinkedLists in code				
PERFORMANCE STANDARD 1.4 : CONSTRUCT SOLUTIONS USING MODULARITY				
1.4.1 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects				
1.4.2 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs				

Content Standard 1.0: Understand Algorithms and Programming

	Unit 1	Unit 2	Unit 3	Unit 4
1.4.3 Construct solutions to problems using student-created components, such as procedures, modules and/or objects				
1.4.4 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution				
1.4.5 Demonstrate code reuse by creating programming solutions using libraries and APIs				
PERFORMANCE STANDARD 1.5 : DEMONSTRATE PROGRAMMING AND DEVELOPMENT				
1.5.1 Systematically design and develop programs for broad audiences by incorporating feedback from users				
1.5.2 Evaluate licenses that limit or restrict the use of computational artifacts when using resources such as libraries				
1.5.3 Evaluate and refine computational artifacts to make them more usable by all and accessible to people with disabilities				
1.5.4 Design and develop computational artifacts while working in team roles and using collaborative tools				
1.5.5 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs				
1.5.6 Plan and develop programs for broad audiences using a software life cycle process				
1.5.7 Explain security issues that might lead to compromised computer programs				
1.5.8 Develop programs for multiple computing platforms				
1.5.9 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project				
1.5.10 Develop and use a series of test cases to verify that a program performs according to its design specifications				
1.5.11 Modify an existing program to add additional functionality and discuss intended and unintended implications, e.g., breaking other functionality				
1.5.12 Evaluate key qualities of a program through a process such as a code review				
1.5.13 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems				