Essentials of Computer Programming

Computer Science Curriculum Framework

2015

Course Title: Essentials of Computer Programming
Course/Unit Credit: 1
Course Number: 460020
Teacher Licensure: Please refer to the Course Code Management System (https://adedata.arkansas.gov/ccms/) for the most current licensure codes.
Grades: 9 - 12
Prerequisite: N/A

Essentials of Computer Programming

Essentials of Computer Programming is a course for students with no formal computer programming experience. Students will spend the majority of time coding to solve real-world problems in a collaborative environment. The course focuses on developing computational thinking, refining problem-solving skills, and applying key programming concepts. Throughout the course, students will use developmentally appropriate and accurate terminology when communicating about technology. Essentials of Computer Programming does not require additional Arkansas Department of Education approval.

Strand	Content Standards
Computational Thinking	
	1. Students will use relevant program design and problem-solving strategies.
	2. Students will evaluate different data representations for storing information.
	3. Students will create and evaluate algorithms to solve problems.
Computing Practice and Programming	
	4. Students will evaluate the use of programming languages to solve problems.
	5. Students will analyze the capabilities of a programming language.
	6. Students will create, test, and use computer programs to solve problems.
Elements of Computing	
	7. Students will classify different types of computers and systems.
	8. Students will describe the major components of a computer system.
	9. Students will analyze the relationships within systems and components.
Social and Ethical Aspects of Programming	
	10. Students will analyze ethical and unethical uses of technology and programming.

Notes:

1. The examples given (e.g.,) are suggestions to guide the instructor.
2. All items in a bulleted list are required to be taught.
3. This curriculum framework is intended to assist in district curriculum development and unit design.
4. This curriculum framework is not a progression document, nor is it intended to be a state-mandated curriculum designating how or when content is taught.
5. The use of the word “language” in this document refers to the programming language selected locally.

Strand: Computational Thinking

Content Standard 1: Students will use relevant program design and problem-solving strategies.

CT.1.ECP.1	Discuss approaches to problem solving used in programming
CT.1.ECP.2	Represent problem-solving logic (e.g., flowcharts, pseudocode, role-play)
CT.1.ECP.3	Interpret logical expressions using Boolean operators (e.g., AND, NOT, OR)
CT.1.ECP.4	Demonstrate operator precedence in expressions (e.g., inside-out, left-to-right, order of operations, right-to-left)
CT.1.ECP.5	Identify the appropriate use of control flow constructs
CT.1.ECP.6	Demonstrate how large problems can be deconstructed into a sequence of smaller problems
CT.1.ECP.7	Analyze and implement collaborative methods in problem solving when appropriate

Strand: Computational Thinking

Content Standard 2: Students will evaluate different data representations for storing information.

CT.2.ECP.1	Classify the types of information that can be stored as variables (e.g., integers, floating points, characters, Booleans, strings)
CT.2.ECP.2	Compare and contrast how variables of different data types are represented in computer memory (e.g., ASCII codes, binary numbers)
CT.2.ECP.3	Compare techniques for manipulating and using data (e.g., searching, sorting)
CT.2.ECP.4	Analyze data organizational structures (e.g., arrays, databases, files)

Key: CT.1.ECP.1 = Computational Thinking.Content Standard 1.Essentials of Computer Programming.1st Student Learning Expectation

Strand: Computational Thinking

Content Standard 3: Students will create and evaluate algorithms to solve problems.

CT.3.ECP.1	Define the term algorithm as used in programming
CT.3.ECP.2	Apply algorithmic reasoning to connect problem-solving processes to programming
CT.3.ECP.3	Investigate ways in which a sequence of algorithms can be combined to create new algorithms
CT.3.ECP.4	Compare and contrast the functional characteristics (e.g., clarity, efficiency, understandability) of algorithms that solve the same problem
CT.3.ECP.5	Develop a procedure to test an algorithm
CT.3.ECP.6	Validate the effectiveness of an algorithm in solving a defined problem

Key: CT.3.ECP.1 = Computational Thinking.Content Standard 3.Essentials of Computer Programming.1st Student Learning Expectation

Strand: Computing Practice and Programming
Content Standard 4: Students will evaluate the use of programming languages to solve problems.

CPP.4.ECP.1	Compare and contrast computer programming paradigms and languages (e.g., machine code, object oriented languages, procedural languages, visual programming languages)
CPP.4.ECP.2	Describe the sequence of steps necessary to create and execute a program on a computer
CPP.4.ECP.3	Manually trace the execution path of code to understand flow of control
CPP.4.ECP.4	Manually trace the execution path of code to determine the output for a sample input

Strand: Computing Practice and Programming
Content Standard 5: Students will analyze the capabilities of a programming language.

CPP.5.ECP.1	Explain how variables are created and initialized
CPP.5.ECP.2	Explain how arrays/lists are created and initialized
CPP.5.ECP.3	Compare and contrast capabilities of a language for conditional branching (e.g., if, if-else, multi-branch)
CPP.5.ECP.4	Compare and contrast capabilities of a language for iteration (e.g., counter controlled, event controlled)
CPP.5.ECP.5	Investigate various capabilities of a language for abstraction including functions, parameters, and return values
CPP.5.ECP.6	Recognize potential problems or limitations with conditional branching, iteration, and abstraction features (e.g., infinite loops, range checking on parameters, unreachable branches)
CPP.5.ECP.7	Analyze traditional programming algorithms (e.g., data management, searching, sorting)
CPP.5.ECP.8	Examine methods used for the input and output of data

Key: CPP.4.ECP.1 = Computing Practice and Programming.Content Standard 4.Essentials of Computer Programming.1st Student Learning Expectation

Strand: Computing Practice and Programming

Content Standard 6: Students will create, test, and use computer programs to solve problems.

CPP.6.ECP.1	Design source code that is efficient and easy to read
CPP.6.ECP.2	Integrate clear and appropriate documentation within student designed source code
CPP.6.ECP.3	Use an editor or visual interface to create and execute programs
CPP.6.ECP.4	Implement computing applications by integrating software development tools and techniques • variables of different data types • arrays/lists of different sizes • conditional branching • iteration • combinations of conditional branching and iteration • functions (e.g., input/output libraries, graphical display libraries, math function libraries, user-defined functions)
CPP.6.ECP.5	Ensure program correctness by utilizing debugging output statements and test cases
CPP.6.ECP.6	Compare and contrast the implementation of two or more programming solutions for the same problem

Key: CPP.6.ECP.1 = Computing Practice and Programming.Content Standard 6.Essentials of Computer Programming.1st Student Learning Expectation

Strand: Elements of Computing

Content Standard 7: Students will classify different types of computers and systems.

EC.7.ECP.1	Identify a variety of electronic devices (e.g., computers, Global Positioning Systems [GPS], smart devices, vehicles) that contain computational processors and execute programs
EC.7.ECP.2	Discuss current trends (e.g., improved user interaction, increasing processor speed and data storage, parallel and distributed systems) in computing

Strand: Elements of Computing

Content Standard 8: Students will describe the major components of a computer system.

EC.8.ECP.1	Describe the hardware components of a computer
EC.8.ECP.2	Identify unique components (e.g., accelerometers, GPS, touch screens) in mobile computing devices
EC.8.ECP.3	Identify the main software components (e.g., operating system, programming tools, user applications) in a computer

Strand: Elements of Computing

Content Standard 9: Students will analyze the relationships within systems and components.

EC.9.ECP.1	Demonstrate an understanding of the relationship of input and output among systems and components
EC.9.ECP.2	Apply strategies to solve routine hardware problems
EC.9.ECP.3	Evaluate tools (e.g., compilers, computer networks, interpreters, operating systems, program editors) necessary to support program execution

Key: EC.7.ECP.1 = Elements of Computing.Content Standard 7.Essentials of Computer Programming.1st Student Learning Expectation

Strand: Social and Ethical Aspects of Programming

Content Standard 10: Students will analyze ethical and unethical uses of technology and programming.

SEA.10.ECP.1	Differentiate among open source, freeware, and proprietary software licenses
SEA.10.ECP.2	Discuss consequences of misuse of information and technology
SEA.10.ECP.3	Discuss social and economic implications (e.g., local, national, global) associated with ethical and unethical hacking and software piracy
SEA.10.ECP.4	Demonstrate legal and ethical behaviors when using information and technology
SEA.10.ECP.5	Apply academic integrity in programming

Key: SEA.10.ECP.1 = Social and Ethical Aspects of Programming.Content Standard 10.Essentials of Computer Programming.1st Student Learning Expectation

Contributors

The following people contributed to the development of this document:

Dr. Keith Bush - University of Arkansas at Little Rock	Andy Hostetler - Jonesboro School District
Dr. Benard Chen - University of Central Arkansas	Tim Johnston - Arkansas Department of Career Education
Gini Cocanower - Bentonville School District	Angela Marsh - University of Arkansas at Monticello
Carl Frank - Arkansas School for Mathematics, Sciences, and the Arts	Mark McDougal - Benton School District
Jordan Frizzell - Star City School District	Anthony Owen - Arkansas Department of Education
Dr. John Gauch - University of Arkansas	Jerry Prince - EAST Initiative
Jimmie Harper - Henderson State University	Jake Qualls - Arkansas State University
Wes Hinesley - Acxiom	Dr. Kenji Yoshigoe - University of Arkansas at Little Rock