Arkansas Computer Science Standards

Coding Block for Grades 7 or 8

2016

Arkansas Computer Science Standards
Coding Block for Grades 7 or 8

Introduction

The computer science 7-8 coding block is designed to be taught during a standalone block of time over a minimum of four to five weeks. As part of this block, students will examine how to formulate algorithms as well as create, analyze, test and debug computer programs in order to solve real-world problems. Students will be required to use a text-based programming language to accomplish these tasks. These standards are not intended to be embedded in activities spread out over multiple courses.

Schools are to ensure that every student receives instruction necessary to meet these standards in either their 7th or 8th grade year. Schools may choose the implementation mechanism that works best for their school and students. Options for implementation include but are not limited to:
- The standards within this block taught as a 4-5 week module within Keyboarding, business elective, or Career Development during the student's 7th or 8th grade year
- The standards within this block taught as a 4-5 week module within another course or specified period of time during the student's 7th or 8th grade year
- The standards within this block taught as part of a high school level programming course for which the school has received approval to offer to 7th or 8th graders

The teacher of record for the Coding Block for Grades 7 or 8 must hold an Arkansas Educator's License in any content area, which allows them to instruct students of the grade level who are taking the block. Though the licensure is open to any content area, it is the responsibility of the school and teacher of record to ensure that the individual providing the instruction has the requisite knowledge needed to teach the block.

Implementation of the Arkansas Computer Science Standards Coding Block for Grades 7 or 8 begins during the 2017-2018 school year.

Computer Science Practices

Students will exhibit proficiency in computer science through:

Perseverance - Students expect and persist in overcoming the challenges that occur when completing tasks. They recognize that making and correcting mistakes will take place during the learning process and problem solving.

Collaboration - Students effectively work and communicate with others ensuring multiple voices are heard and considered. They understand that diverse thoughts may lead to creative solutions and that some problems may be best solved collaboratively.

Patterns - Students understand and utilize the logical structure of information through identifying patterns and creating conceptual models. They decompose complex problems into simpler modules and patterns.

Tools - Students evaluate and select tools to be used when completing tasks and solving problems. They understand that appropriate tools may include, but are not limited to, their mind, pencil and paper, manipulatives, software application programs, programming languages, or appropriate computing devices.

Communication - Students effectively communicate, using accurate and appropriate terminology, when explaining the task completion or problem solving strategies that were used. They recognize that good documentation is an ongoing part of the process, and when appropriate provide accurate documentation of their work in a manner that is understandable to others.

Ethics and Impact - Students comprehend the ramifications of actions prior to taking them. They are aware of their own digital and cyber presence and its impact on other individuals and society.

Problem Solving - Students exhibit proficiency in Computer Science through identifying and systematically solving problems (e.g., engineering design process). They recognize problem solving as an ongoing process.

Strand	Content Cluster
Computational Thinking and Problem Solving	
	1. Students will examine and formulate algorithms that solve specific problems.
Algorithms and Programs	
	2. Students will create programs that solve problems.
	3. Students will analyze, test, and debug computer programs.

Notes for the Computer Science Standards document:
1. The examples given (e.g.,) are suggestions to guide the instructor.
2. The Practices are intended to be habits of mind for all students and were written broadly in order to apply to all grades. The Practices are not content standards and are not intended to be formally assessed but may be assessed formatively.
3. This Arkansas Department of Education curriculum standards document is intended to assist in district curriculum development, unit design, and to provide a uniform, comprehensive guide for instruction.
4. Notes found within the document are not approved by the Arkansas State Board of Education, but are provided for clarification of the standards by the Arkansas Department of Education and/or the standards drafting committee. The notes are subject to change as understandings of the standards evolve.

Strand: Computational Thinking and Problem Solving
    Content Cluster 1: Students will examine and formulate algorithms that solve specific problems.

CT.1.B.1	Examine traditional programming algorithms, including searches and sorts
CT.1.B.2	Describe the steps needed to efficiently solve a problem
CT.1.B.3	Manually test algorithms with sample data to observe accuracy of anticipated output

Strand: Algorithms and Programs
    Content Cluster 2: Students will create programs that solve problems.

A.2.B.1	Implement the following programming concepts: • data types • variable declaration and initialization • assignment statements • conditional branching (e.g., if, if-else, multi-branch) • iteration (e.g., while, do-while, for)
A.2.B.2	Create a program using a text-based programming language

Strand: Algorithms and Programs
    Content Cluster 3: Students will analyze, test, and debug computer programs.

A.3.B.1	Explain the logic involved in how a computer program executes
A.3.B.2	Utilize basic documentation/comments (e.g., program headers, pseudocode)
A.3.B.3	Test a computer program with data and evaluate output for accuracy
A.3.B.4	Find and debug errors in a computer program

Contributors

The following people contributed to the development of this document:

Melinda Bailey – Alma School District	Tim Johnston – Arkansas Department of Career Education
Shena Ball – Mid-America Science Museum	Glenda Kennedy – University of Arkansas at Pine Bluff
Christina Bishop – Hot Springs Public School District	Lisa Kraus – Horatio Public School District
Marjo Burk – Fayetteville Public School District	Mickey McFetridge – Springdale Public School District
Tina Calhoon – Arkadelphia Public School District	Mike McMillan – Pulaski Technical College
Shari Cates – North Little Rock School District	Carolyn McNeely – Crowley's Ridge Educational Service Cooperative
G.B. Cazes – Cyber Innovation Center	David Nance – Arkansas Department of Education
Krystal Corbett – Cyber Innovation Center	Heather Newsam – Arkansas Department of Education
Becky Cunningham – Arkansas Tech University	Anthony Owen – Arkansas Department of Education
Stacy DeFoor – Conway Public School District	Jerry Prince – EAST Initiative
Donna Dyson – Hamburg Public School District	Jake Qualls – Arkansas State University
Dan Eddy – Bentonville Public School District	Stacy Reynolds – Star City Public School District
Carl Frank – Arkansas School for Mathematics, Sciences and the Arts	Karl Romain – Little Rock School District
Marianne Hauser – Fayetteville Public School District	Deborah Seehorn – Computer Science Teachers Association
Haihong Hu – University of Central Arkansas	Nicholas Seward – Arkansas School for Mathematics, Sciences and the Arts
Ronda Hughes – Hot Springs Public School District	Joel Spencer – Little Rock School District