Arkansas Computer Science Standards for Grades 9-12

Advanced Programming

2016

Advanced Programming

Introduction

The Arkansas Advanced Programming Course focuses on the skills necessary to design and develop reliable programs and software. Through these standards, students will explore, apply, and advance toward mastery of data structures and algorithms, data representation, algorithm design, and program efficiency. Students will accomplish tasks and solve problems independently and collaboratively with the tools and skills needed to be successful in college and careers.

The Arkansas State Board of Education (SBE) does not place any pre-requisites on the Arkansas Computer Science High School Courses, but allows for schools to place students in any of the courses based on ability and desire. The Arkansas Department of Education (ADE) recommends that districts develop and formally adopt a written policy outlining placement protocols. Evaluation tools and placement criteria will be the responsibility of the local districts. Though there are no specific course prerequisites, students enrolling in Advanced Programming, Advanced Networking, or Advanced Information Security should understand and be able to apply the content/concepts found within the Arkansas Computer Science Courses Levels 1 - 4.

The SBE and ADE authorizes schools to enroll students across levels and emphases in the same sections of the master schedule (a.k.a. stacking) as long as the number of students does not exceed Standards of Accreditation maximums and/or ratios, and the school can reasonably assure a high-quality educational experience for all students within that section.

Implementation of the Arkansas Computer Science Standards for Grades 9-12 begins during the 2017-2018 school year.

	Advanced Programming Level 1	**Advanced Programming Level 2**
Advanced Programming	465050	465060
Game Design	465650	465660

Course Title: Advanced Programming Levels 1 & 2
Course/Unit Credit: 0.5 Credits per Course/level

Teacher Licensure: Please refer to the Course Code Management System (https://adedata.arkansas.gov/ccms/) for the most current licensure codes.
Grades: 9-12
Prerequisites: There are no ADE established course prerequisites for any of the Computer Science levels; it is up to the local district to determine placement based on student ability.

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.

Advanced Programming

Strand	Content Cluster
Computational Thinking and Problem Solving	
	1. Students will analyze problem-solving strategies.
	2. Students will solve problems cooperatively and collaboratively.
Data and Information	
	3. Students will analyze various ways in which data is represented.
	4. Students will collect, arrange, and represent data.
	5. Students will interpret and analyze data and information.
Algorithms and Programs	
	6. Students will create, evaluate, and modify algorithms.
	7. Students will create programs to solve problems.
Computers and Communications	
	8. Students will analyze various components and functions of computers.
Community, Global, and Ethical Impacts	
	9. Students will analyze appropriate uses of technology.

Notes for the Computer Science Standards for High School 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 analyze problem-solving strategies.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.1.1 Judge the effectiveness of several different problem-solving strategies (e.g., look for a pattern, try a simpler case, work backwards) and determine the most effective solution for a given problem	APL2.1.1 *Continuation of this standard is not specifically included or excluded*
APL1.1.2 Utilize an abstract model (e.g., computational model, computer model) to simulate a system by reproducing behavior of a system	APL2.1.2 *Continuation of this standard is not specifically included or excluded*

Strand: Computational Thinking and Problem Solving

Content Cluster 2: Students will solve problems cooperatively and collaboratively.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.2.1 Utilize an appropriate development life cycle process (e.g., spiral, waterfall) as a member of a development team for a given project (e.g., community service project, real world) of level-appropriate complexity	APL2.2.1 Utilize an appropriate development life cycle process (e.g., spiral, waterfall) while building a development team for a given project (e.g., community service project, real world) of level-appropriate complexity
APL1.2.2 Observe global collaboration in the development of a computational artifact (e.g., review the process through which an open-source software project hosted on GitHub has improved over time)	APL2.2.2 Contribute to global collaboration in the development of a computational artifact (e.g., assist in resolving a bug in an open-source software project hosted on GitHub)

Strand: Data and Information

Content Cluster 3: Students will analyze various ways in which data is represented.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.3.1 Compare the differences in ways level-appropriate data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) organize data	APL2.3.1 Compare the differences in ways level-appropriate data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) organize data
APL1.3.2 *This standard is not specifically required until Level 2*	APL2.3.2 Compare media formats (e.g., graphics, sounds) for traits such as lossiness and compression performance

Strand: Data and Information

Content Cluster 4: Students will collect, arrange, and represent data.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.4.1 Select and use data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) based on functionality, storage, and performance tradeoffs	APL2.4.1 Implement data structures (e.g., graphs, linked lists, maps, queues, sets, stacks, trees) to support the creation of larger computational artifacts
APL1.4.2 Create and populate tables in databases	APL2.4.2 *Continuation of this standard is not specifically included or excluded*
APL1.4.3 *This standard is not specifically required until Level 2*	APL2.4.3 Use various data collection techniques for different types of problems (e.g., system sensors, mobile device GPS, open data sets, social media data sets, user surveys)

Strand: Data and Information

Content Cluster 5: Students will interpret and analyze data and information.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.5.1 Issue queries against databases to glean meaning from stored data	APL2.5.1 Discuss real-world data sources that can be mined to produce new knowledge

Strand: Algorithms and Programs

Content Cluster 6: Students will create, evaluate, and modify algorithms.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.6.1 Evaluate multiple classical algorithms in terms of time and space complexities (e.g., Big O notation)	APL2.6.1 Evaluate multiple student-created algorithms in terms of time and space complexities (e.g., Big O notation)
APL1.6.2 Develop algorithms to solve student-identified problems of appropriate complexity	APL2.6.2 Develop algorithms to solve student-identified problems of appropriate complexity
APL1.6.3 Decompose problems of appropriate complexity into well-defined steps to produce computational artifacts	APL2.6.3 Decompose problems of appropriate complexity into well-defined steps to produce computational artifacts

Strand: Algorithms and Programs

Content Cluster 7: Students will create programs to solve problems.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.7.1 Demonstrate code reuse by creating programming solutions using level-appropriate libraries and application program interfaces (API) (e.g., graphics libraries, maps)	APL2.7.1 Demonstrate code reuse by creating programming solutions using level-appropriate libraries and APIs (e.g., graphics libraries, maps)
APL1.7.2 *This standard is not specifically required until Level 2*	APL2.7.2 Break a large system down into progressively smaller classes or objects that are responsible for some part of the problem domain
APL1.7.3 Create programs to solve problems of level-appropriate complexity	APL2.7.3 Create programs to solve problems of level-appropriate complexity

Strand: Computers and Communications

Content Cluster 8: Students will analyze various components and functions of computers.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.8.1 Investigate the functionality of various hardware components (e.g., game controllers, input and output devices, robotics components, sensors)	APL2.8.1 Integrate various hardware components (e.g., game controllers, input and output devices, robotics components, sensors) as they relate to student-developed computational artifacts

Strand: Community, Global, and Ethical Impacts

Content Cluster 9: Students will analyze appropriate uses of technology.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS	
Level 1	Level 2
APL1.9.1 Investigate the role and impact of laws and regulations on the development and use of software (e.g., privacy, security)	APL2.9.1 Clarify the role and impact of laws and ethical decisions of student-led projects
APL1.9.2 Evaluate security issues that lead to compromised computer programs (e.g., circular references, lack of error checking and field size checking)	APL2.9.2 Implement security policies by comparing encryption and authentication strategies (e.g., safeguarding keys, secure coding)
APL1.9.3 Use an ethical decision-making process to justify decisions made in creating computational artifacts	APL2.9.3 Defend an ethical decision-making process to justify decisions made in creating computational artifacts

Appendix 1: Advanced Programming: Game Design

This appendix contains exceptions that apply to the teaching of Game Design under the High School Computer Science Advanced Programming standards. Students enrolled in Computer Science Game Design at any level, must receive instruction in all High School Computer Science Advanced Programming Standards within the Advanced Programming level to which the Game Design course appends. The following exceptions apply to the standard indicated and modify the requirements of that standard only, all other standards within that level must be taught as presented above, and any additional standards specific to Game Design will be listed at the end of the exceptions.

High School Computer Science Advanced Programming Level 1: Game Design Level 1	
APL1.1.2	To meet this standard, include physics and mathematics principles for game mechanics.
APL1.7.1	To meet this standard, include the function of a game engine and supporting libraries.
APL1.8.1	Investigate the functionality of various hardware components as they relate to modern game design (e.g., game controllers, GPU acceleration, input and output devices such as sensors, robotics components, virtual reality).
APL1.9.1	To meet this standard, include game rating system (ERSB) and copyright laws concerning assets.
APL1.10.1	Compare and contrast game elements present in board games and digital representations.
APL1.10.2	Describe how the parts of a game contribute to its overall behavior (e.g., addiction, balance, character, enjoyment factor, goal, immersion, interactive, level, movement, replay value, rules, score, story, theme, victory/goal conditions).
APL1.10.3	Describe the core areas of digital game design: • Characters and Development • Core Mechanics • Gameplay Modes • Game World • Level Design • Mode Elaboration • Story Elaboration
APL1.10.4	Describe the concept of a game loop in digital games.
APL1.10.5	Describe the functions of a game engine and supporting libraries (eg. images, sounds, sprites, text effects)
APL1.10.6	Discuss common asset creation techniques (e.g., images, music, sounds, 3-D models)
APL1.10.7	Classify the roles and responsibilities of each member on a game design team • Artist/Animation • Audio • Designer • Producer • Programmers • Project manager • Quality Assurance

APL1.10.8	Summarize the history of games and their significance to digital games
APL1.10.9	Summarize the role of play in human culture
High School Computer Science Advanced Programming Level 2: Game Design Level 2	
APL2.7.1	To meet this standard, include the function of a game engine and supporting libraries.
APL2.10.1	Design a game following the core areas of digital design (e.g. level maps, rules, script writing, storyboarding, storytelling).
APL2.10.2	Use concepts related to updating and drawing within the game loop.
APL2.10.3	Utilize custom assets for a game.
APL2.10.4	Discuss diverse careers that are available for game designers and their educational requirements.
APL2.10.5	Identify the hard and soft skills required to be an effective game design team member (e.g., analytical competence, creativity, initiative, punctuality, teamwork).
APL2.10.6	Explore advancing and emerging technologies (e.g., artificially intelligent agents, robotics) as they relate to game design.

Contributors

The following people contributed to the development of this document:

Stephany Alhajjaj – Little Rock School District	Lori Kagebein – Wonderview School District
Jeff Anderson – Rogers Public Schools	Jeff Matocha – Ouachita Baptist University
Brent Burgin – Dassault Falcon Jet	Daniel Moix – Arkansas School for Mathematics, Sciences, and the Arts
Kristian Cartwright – Fayetteville Public Schools	Larry Morell – Arkansas Tech University
Kevin Collins – Alma School District	David Nance – Arkansas Department of Education
Cecil Cossey – Hamburg School District	Thad Nipp – Alma School District
Ty Davis – Springdale Public Schools	Anthony Owen – Arkansas Department of Education
Jennifer Feltmann – Berryville Public Schools	Kenneth Powell – Metova Federal
Carl Frank – Arkansas School for Mathematics, Sciences, and the Arts	Jerry Prince – EAST Initiative
Charles Gardner – Cyber Innovation Center	Kimberly Raup – Conway Public Schools
Tammy Glass – Spring Hill School District	Sandra Rhone – Mineral Springs School District
Tommy Gober – Cyber Innovation Center	Linda Riley – Wonderview School District
Joel Gordon – Arkansas Regional Innovation Hub	Nicholas Seward – Arkansas School for Mathematics, Sciences, and the Arts
Marilyn Harris – Virtual Arkansas	Tom Simmons – El Dorado Public Schools
Andy Hostetler – Jonesboro Public Schools	Dustin Summey – Virtual Arkansas
Tim Johnston – Arkansas Department of Career Education	Travis Taylor – Little Rock School District
Linda Joplin – Fort Smith Public Schools	Karma Turner – Lake Hamilton School District