Arkansas CTE

Introduction to Computer Science (2025)

Careers and Professionalism

A Standard 1.1 - Demonstrate professional readiness within the computer science domain. I.A

1. 1.1.1 Demonstrate understanding of various career paths in computer science (e.g., software development, data science, cybersecurity, networking, robotics, computer engineering, and artificial intelligence) and their roles in different industries. I.A.1
2. 1.1.2 Practice professional communication skills through various activities (e.g., mock interviews, technical presentations, and collaborative coding projects) to prepare for real-world scenarios in computer science careers. I.A.2
3. 1.1.3 Describe the importance of ethical considerations in computer science (e.g., data privacy, algorithmic bias, and social impact of technology) and how they apply to professional practice. I.A.3
4. 1.1.4 Evaluate the concept and importance of a professional online presence in computer science careers (e.g., portfolio websites, professional networking platforms, open-source contributions) and explore ways to showcase skills and projects. I.A.4
5. 1.1.5 Identify key elements of a professional portfolio and explain the purpose of a portfolio (e.g., showcasing skills, documenting growth, and supporting college applications or job searches). I.A.5

Computational Thinking

A Standard 2.1 - Apply computational thinking to effectively solve real-world problems. II.A

1. 2.1.1 Recognize situations where leveraging computational approaches (e.g., data analysis, automation, or simulations) would be beneficial for solving real-world problems. II.A.1
2. 2.1.2 Apply the core computational thinking principles—abstraction, decomposition, algorithm development, and pattern recognition—to designing effective problem-solving strategies. II.A.2

B **Standard 2.2 - Access, manipulate, and store data within a program using a text-based programming language.** II.B

1. 2.2.1 Use various data types (e.g., Booleans, characters, integers, floating points, and strings) appropriately within a program. II.B.1
2. 2.2.2 Create and use variables to store and manage data within a program. II.B.2
3. 2.2.3 Construct expressions using arithmetic operators (e.g., +, -, *, /, and %) and numeric data types to perform calculations within a program. II.B.3
4. 2.2.4 Convert between different data types when necessary within a program (e.g., casting a string into an integer). II.B.4
5. 2.2.5 Perform operations that encode and decode data from one form into another form (e.g., binary to hexadecimal, numeric values to colored pixels, or numbers to ASCII/Unicode representations). II.B.5
6. 2.2.6 Implement data structures (e.g., arrays, lists, sets, and maps) to organize, store, manipulate, and perform operations on collections of data within a program. II.B.6

C **Standard 2.3 - Design and create programs by understanding the execution sequence and anticipated outputs using a text-based programming language.** II.C

1. 2.3.1 Analyze a program in terms of steps of execution and expected outcomes (e.g., storyboards, flowcharts, and pseudocode). II.C.1
2. 2.3.2 Construct Boolean expressions using relational operators (e.g., <, >, <=, >=, ==, and !=) within a program. II.C.2
3. 2.3.3 Construct Boolean expressions using logical operators (e.g., AND, OR, and NOT) within a program. II.C.3
4. 2.3.4 Create programs that implement selection control structures (e.g., if statements and switch statements) to make decisions and execute different code paths based on conditions. II.C.4
5. 2.3.5 Create programs that implement iteration control structures (e.g., while loops and for loops) to repeat code blocks a specific number of times or until a condition is met. II.C.5
6. 2.3.6 Create subroutines (e.g., procedures and functions) to modularize code for reusability and organization within a program. II.C.6
7. 2.3.7 Debug errors (e.g., syntax, runtime, and logic) within a program to ensure program functionality. II.C.7

D **Standard 2.4 - Transfer data into and out of a program effectively using a text-based programming language.** II.D

1. 2.4.1 Use the console for basic data input and output operations within a program. II.D.1
2. 2.4.2 Explain the structure and purpose of different file types (e.g., txt, csv, bmp, and json) used for data storage. II.D.2
3. 2.4.3 Develop programs that perform file operations including reading data from, writing data to, and appending data to files. II.D.3

E **Standard 2.5 - Develop programs that adhere to good coding conventions and include clear documentation.** II.E

1. 2.5.1 Implement consistent formatting and naming conventions (e.g., indentation, spacing, variable names) to improve code readability and maintainability. II.E.1
2. 2.5.2 Craft clear and concise comments within the code to explain the purpose of different code sections, algorithms used, and non-obvious logic. II.E.2

Data Analysis and Visualization

A **Standard 3.1 - Investigate different types of data and explore various methods for collecting data.** III.A

1. 3.1.1 Categorize data into different types (e.g., quantitative vs. qualitative) and understand the distinction between them. III.A.1
2. 3.1.2 Identify potential sources of data (e.g., sensors, surveys, databases, and web scraping) based on the type of data needed. III.A.2
3. 3.1.3 Explain advantages and disadvantages of different data collection methods considering accuracy, cost, and time. III.A.3
4. 3.1.4 Design basic data collection methods (e.g., surveys, checklists, experiments) for a specific purpose. III.A.4

B **Standard 3.2 - Develop programs that effectively communicate data insights using analysis and visualizations.** III.B

1. 3.2.1 Identify and address data quality issues (e.g., missing values, inconsistencies, and outliers). III.B.1
2. 3.2.2 Analyze data sets using descriptive statistics and visualizations (e.g., histograms and box plots). III.B.2
3. 3.2.3 Inspect and use exploratory analysis techniques to identify patterns and relationships. III.B.3
4. 3.2.4 Develop programs that perform appropriate data analysis techniques (e.g., correlations, comparing means). III.B.4
5. 3.2.5 Create programs that generate visualizations (e.g., bar, line, scatter, pie charts). III.B.5

C **Standard 3.3 - Communicate data insights effectively, considering data bias and ethics.** III.C

1 3.3.1 Recognize and explain how potential biases in data can influence insights. III.C.1

2 3.3.2 Develop narratives that communicate data insights clearly and effectively. III.C.2

3 3.3.3 Demonstrate understanding of correlation vs. causation. III.C.3

4 3.3.4 Develop sound inferences to support informed decision-making. III.C.4

5 3.3.5 Present positive and negative findings in an unbiased way. III.C.5

Major and Emerging Technologies

A **Standard 4.1 - Develop understanding of major established technologies and their impact.** IV.A

1 4.1.1 Define artificial intelligence (AI), its subfields, benefits, and drawbacks. IV.A.1

2 4.1.2 Explain cloud computing, its service models, and its impact on IT. IV.A.2

3 4.1.3 Define the Internet of Things (IoT), its applications, and its impact. IV.A.3

4 4.1.4 Explain big data, its characteristics, and its impact on industry. IV.A.4

B **Standard 4.2 - Investigate emerging technologies and their potential impact.** IV.B

1 4.2.1 Define quantum computing and explain its potential to transform computing. IV.B.1

2 4.2.2 Describe edge computing and its advantages in processing data. IV.B.2

3 4.2.3 Explain extended reality (VR, AR, MR) and its applications. IV.B.3

4 4.2.4 Describe blockchain technology and its potential uses in various sectors. IV.B.4

Cybersecurity, Computer Systems, and Networking

A **Standard 5.1 - Understand the importance of cybersecurity and its role in protecting information.** V.A

1 5.1.1 Analyze core information security principles (confidentiality, integrity, availability, authentication, non-repudiation). V.A.1

2 5.1.2 Describe common cyber threats (e.g., malware, phishing, ransomware). V.A.2

3 5.1.3 Describe cybersecurity practices and technologies (e.g., encryption, firewalls, MFA). V.A.3

4 5.1.4 Analyze the impact of best practices (e.g., strong passwords, awareness training). V.A.4

B **Standard 5.2 - Identify and explain core components of computer systems.** V.B

1. 5.2.1 Identify and explain key hardware components and their purposes. V.B.1
2. 5.2.2 Define software and distinguish between system and application software. V.B.2
3. 5.2.3 Explain basic operating system functions and their importance. V.B.3
4. 5.2.4 Summarize open-source software and its impact. V.B.4
5. 5.2.5 Explain the importance of software updates and patches. V.B.5

C **Standard 5.3 - Identify and explain core concepts of computer networks.** V.C

1. 5.3.1 Define and explain common network concepts (e.g., IP, protocols, latency). V.C.1
2. 5.3.2 Identify and describe functionalities of common network hardware. V.C.2
3. 5.3.3 Explain basic network security principles used to safeguard networks. V.C.3
4. 5.3.4 Analyze the relationship between computer systems, software, and networks. V.C.4