COMPETENCIES FOR SECONDARY TEACHERS: COMPUTER SCIENCE, 4-12

In addition to the Arkansas Teaching Standards (ATS) and the competencies for the Teacher Excellence and Support System (TESS), including competencies regarding the knowledge and use of educational technology that reflect the International Society for Technology in Education standards, the teacher of Computer Science, grades 4-12, shall also demonstrate knowledge and competencies in the following areas:

1. COMPUTING SYSTEMS

K-12 Computer Science Framework

CS for Arkansas

Praxis 5652: Section V

1.1 Know that operating systems are programs that control and coordinate interactions between hardware and software components
- Identify hardware components and their functions
- Identify software components and their functions
- Identify common operating systems tasks
- Identify resource issues that have an impact on functionality

1.2 Be familiar with computing systems embedded in everyday objects (e.g., Internet of Things (IoT), ATMs, medical devices)
- Describe what an embedded system is
- Define what the IoT is and how it is used
- Describe how sensors are used in embedded systems

1.3 Be familiar with computers as layers of abstraction from hardware (e.g., logic gates, chips) to software (e.g., system software, applications)
- Identify appropriate abstraction layers for hardware and software components

1.4 Be familiar with the steps required to execute a computer program (e.g., fetch-decode-execute cycles)
- Describe what happens during fetch, decode, and execute, including the order of the steps in the cycle

1.5 Know the capabilities, features, and uses of different types of computing systems (e.g., desktop, mobile, cluster)
- Identify capabilities, features, and uses for each type of computer system
- Identify criteria to evaluate and compare computing systems

2. NETWORKS AND THE INTERNET

K-12 Computer Science Framework

CS for Arkansas

Praxis 5652: Section V

2.1 Be familiar with trade-offs between local, network, and cloud computing and storage
- Identify advantages and disadvantages in terms of performance, cost, security, reliability, and collaboration
- Identify means of storing binary data

2.2 Be familiar with communication between devices
- Identify and compare wireless communication systems
- Identify and compare wired communication systems
- Identify and compare network types

2.3 Know components of networks
- Identify network hardware devices and their functions
- Describe possible abstraction models of networks

2.4 Be familiar with factors that have an impact on network functionality

COMPETENCIES FOR SECONDARY TEACHERS: COMPUTER SCIENCE, 4-12

- Define basic terminology (e.g., bandwidth, load, latency)
- Estimate necessary bandwidth and data size for a given situation
- Identify critical resources for a given situation

2.5 Be familiar with how Internet and Web protocols work

- Describe the purpose of protocols and identify common Internet and Web protocols
- Compare IPv4 and IPv6
- Identify and describe the basic parts of a URL (e.g., protocol, subdomain, domain name, port, path)
- Describe the hierarchical structure of names in the domain name system (DNS)
- Describe the purpose and function of IP addressing
- Identify how Internet protocols address reliability, redundancy, and error handling

2.6 Be familiar with digital and physical strategies for maintaining security

- Identify characteristics of strong passwords (e.g., length, bits per character)
- Identify digital and physical security strategies
- Identify trade-offs in the use of security measures (e.g., encryption, description, digital signatures and certificates)

2.7 Be familiar with concepts of cybersecurity

- Identify and define the five pillars of cybersecurity: confidentiality, integrity, availability, nonrepudiation, and authentication

2.8 Be familiar with the components that make up the Web (e.g., HTTP, HTML, browsers, servers, clients)

- Identify the uses of markup languages
- Identify the purposes of browsers, servers, and clients

3. DATA AND ANALYSIS

K-12 Computer Science Framework

CS for Arkansas

Praxis 5652: Section IV

3.1 Understand bits as the universal medium for expressing digital information

- Perform calculations, using bits and bytes
- Determine the number of bits and bytes required to store a given amount of data
- Given the description of an encoding scheme, encode or decode data
- Describe lossy and lossless data compression
- Explain why binary numbers are fundamental to the operation of computer systems

3.2 Be familiar with concepts of data encryption and decryption

- Distinguish between encoding and encryption
- Identify trade-offs in the use of data encryption

COMPETENCIES FOR SECONDARY TEACHERS: COMPUTER SCIENCE, 4-12

3.3 Know how to use computational tools, including spreadsheets, to analyze data in order to discover, explain, and visualize patterns, connections, and trends

- Transform data to make it more useful
- Identify specific data or characteristics of specific data that need to be removed or modified before an entire data set can be used
- Describe the use of spreadsheet operations (e.g., formulas, filters, sorts, charts, graphs) to analyze and visualize data

3.4 Be familiar with the use of computing in simulation and modeling

- Describe questions that can be answered with a given simulation, or explain what data and process are required in a simulation in order to answer a given question
- Trace code in a simulation context
- Identify missing code in a simulation context
- Identify the impact of changes to simulations (e.g., more or fewer variables, more or less data)
- Identify applications of simulation and modeling

3.5 Be familiar with methods to store, manage, and manipulate data

- Use terminology and concepts of files and databases
- Identify measures of file size (e.g., byte, kilo, mega, giga, tera, peta)
- Identify issues connected with the storage requirements of computing applications, including scale, redundancy, and backup

3.6 Be familiar with a variety of computational methods for data collection, aggregation, and generation

- Identify the benefits of working with publicly available data sets
- Identify the types of data generated by surveys and sensors
- Identify examples of crowdsourcing and citizen science
- Identify appropriate data-collection methods for a given context and purpose

4. ALGORITHMS AND PROGRAMMING

K-12 Computer Science Framework

CS for Arkansas

Praxis 5652: Section II and III

4.1 Understand and apply knowledge of abstraction, pattern recognition, problem decomposition, number base conversion, and algorithm formats

- Understand abstraction as a foundation of computer science
- Know how to use pattern recognition, problem decomposition, and abstraction to develop an algorithm
- Understand number base conversion and binary, decimal, and hexadecimal number systems
- Understand how to develop and analyze algorithms expressed in multiple formats (e.g., natural language, flowcharts, pseudocode)

COMPETENCIES FOR SECONDARY TEACHERS: COMPUTER SCIENCE, 4-12

4.2 Understand and apply knowledge of algorithms, recursive algorithms, and randomization

- Be familiar with the limitations of computing in terms of time, space, and solvability as well as with the use of heuristic solutions that can address these limitations
- Understand searching and sorting algorithms; can analyze sorting algorithms for correctness and can analyze searching algorithms for correctness and efficiency
- Understand simple recursive algorithms (e.g., n factorial, sum of first n integers)
- Be familiar with the use of randomization in computing

4.3 Understand and apply knowledge of programming control structures, standard operators, variables, correctness, extensibility, modifiability, and reusability

- Understand how to write and modify computer programs in a text-based programming language
- Understand how to analyze computer programs in terms of correctness
- Know the concepts of extensibility, modifiability, and reusability
- Understand the three basic constructs used in programming: sequence, selection, and iteration
- Understand how to use standard operators (i.e., assignment, arithmetic, relational, logical) and operator precedence to write programs
- Understand how to use variables and a variety of data types

4.4 Understand and apply knowledge of procedures, event-driven programs, usability, data structures, debugging, documenting and reviewing code, libraries and APIs, IDEs, and programming language paradigms, including object-oriented concepts

- Understand how to write and call procedures with parameters and return values
- Know the concepts of event-driven programs that respond to external events (e.g., sensors, messages, clicks)
- Be familiar with usability and use experience (e.g., ease of use and accessibility)
- Be familiar with dictionaries/maps, stacks, and queues
- Understand how to use debugging techniques and appropriate test cases
- Be familiar with characteristics of well-documented computer programs that are usable, readable, and modular
- Be familiar with techniques to obtain and use feedback to produce high-quality code (e.g., code reviews, peer feedback, end user feedback)
- Know how to use libraries and APIs
- Understand programming techniques to validate correct input and detect incorrect input
- Be familiar with the features and capabilities of integrated development environments (IDEs)

COMPETENCIES FOR SECONDARY TEACHERS: COMPUTER SCIENCE, 4-12

- Be familiar with the differences between low- and high-level programming languages
- Be familiar with different programming paradigms
- Know object-oriented programming concepts
- Be familiar with program compilation and program interpretation

5. IMPACTS OF COMPUTING

K-12 Computer Science Framework

CS for Arkansas

Praxis 5652: Section I

5.1 Understand computing as a way of expressing creativity, solving problems, enabling communication, and fostering innovation in a variety of fields and careers

- Recognize that computers can be used to showcase creativity
- Recognize the benefits of using computers to solve problems
- Provide examples of how computers enable communication and collaboration
- Provide examples of how computers foster innovation

5.2 Know the obstacles to equal access to computing among different groups and the impact of those obstacles

- Identify obstacles to equal access to computing among different groups (e.g., groups defined by gender, socioeconomic status, disability/accessibility needs) and the impact of those obstacles
- Identify factors that contribute to the digital divide
- Match obstacles to equal access with effective solutions

5.3 Understand beneficial and harmful effects of computing innovations and the trade-offs between them

- Analyze computing innovations in terms of their social, economic, and cultural impacts, both beneficial and harmful
- Identify trade-offs between beneficial and harmful effects of computer innovations

5.4 Know different methods of protecting intellectual property rights and the trade-offs between them in a variety of contexts (e.g., Creative Commons, open source, copyright)

- Using correct vocabulary, describe how different methods of protecting intellectual property rights work
- Given a context, identify appropriate methods of protecting intellectual property rights
- Identify and compare trade-offs between different methods of protecting intellectual property rights

- 5.5 Understand ethical and unethical computing practices and their social, economic, and cultural implicationsIdentify ethical and unethical computing practices in context
  - Describe the social, economic, and cultural implications of ethical and unethical computing practices
  - Identify the conditions under which a given computing practice is ethical or legal

COMPETENCIES FOR SECONDARY TEACHERS: COMPUTER SCIENCE, 4-12

5.6 Know privacy and security issues regarding the acquisition, use, and disclosure of information in a digital world

- Using correct vocabulary, describe privacy and security issues
- In context, identify appropriate strategies to safeguard privacy and ensure security
- Describe trade-offs between local and cloud-based data storage
- Identify methods that digital services use to collect information about users

6. COMPUTATIONAL ARTIFACTS

K-12 Computer Science Framework

CS for Arkansas

6.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations
6.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue
6.3 Modify an existing artifact to improve or customize it
6.4 Systematically test computational artifacts by considering all scenarios and using test cases
6.5 Identify and fix errors using a systematic process
6.6 Evaluate and refine a computational artifact multiple times to enhance its performance reliability, usability, and accessibility
6.7 Utilize bleeding edge technologies to create, develop, and present computational artifacts

7. EMERGING TRENDS

CS for Arkansas

7.1 Be familiar with emerging computing technologies (e.g., Internet of Nano Things, virtual reality, augmented reality)
7.2 Be familiar with emerging networking technologies (e.g., smart grid communications, cloud computing, wireless sensor networks, wireless protocols, data transmission rates)
7.3 Be familiar with emerging uses of data and methods of data analysis (e.g., artificial intelligence, neural networks, speech and image recognition, problem solving)
7.4 Be familiar with emerging procedures and capabilities of programming (e.g., blockchain cryptocurrencies, blockchain ledger systems, artificial intelligence, Rust, Google Go, Swift, Hack, Julia)
7.5 Be familiar with emerging impacts of computing (e.g., effects of robotics in manufacturing, ethical considerations of artificial intelligence, privacy issues with emerging technologies)

8. PEDAGOGY AND STANDARDS

CSTA Standards
ISTE Standards
K-12 Computer Science Framework
CS for Arkansas
AP Computer Science Principles

8.1 Be familiar with best practices for teaching computer science concepts
8.2 Be familiar with current computer science standards for educators and students, such as:

- Computer Science Teachers Association Standards (CSTA)
- International Society for Technology in Education Standards (ISTE)
- K-12 Computer Science Framework
- Arkansas Computer Science Standards and courses
- Advanced Placement Computer Science courses