5440-14 Computer Science Educator

1. *The holder is authorized to teach computer science in grades 7-12.*

2. In order to qualify for this endorsement, the candidate shall demonstrate the following:

3. Knowledge Standards:

3.1. **Demonstrates knowledge of essential computer science concepts and skills, including:**

3.2. Historical Context~~Program and algorithm design; data structures; object-oriented program design; and high-level languages~~

3.2.1. Important contributions of individuals or groups, particularly those made by underrepresented populations', to the development of computer technology

3.2.2. Generational milestones in the historical development of computer technology

3.3. Algorithmic Thinking~~Computer hardware architecture~~

3.3.1. The basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solutions, testing, evaluation, revising).

3.4. Computing Systems~~Programming languages, including the definition and structure of languages and comparison of existing high-level languages~~

3.4.1. The function, application, capabilities and limitations of computers, their operating systems, software applications, and networking components

3.4.2. Appropriate use of hardware components (e.g. input, processing, output, primary / secondary storage devices) with respect to functionality, cost, size, speed, accessibility, and aesthetics

3.4.3. The role of compilers and interpreters in translating programming languages into machine instructions

3.5. Networks and The Internet~~Fluency in at least two high-level languages used in current pedagogy~~

3.5.1. Various types of networks and their performance characteristics, models for defining network standards and protocols, and network topology

3.5.2. Cybersecurity including identifying features and functions of security tools (e.g., firewalls, antivirus programs, filtering software, encryption).

3.5.3. The relationship between clients and servers on a network (e.g, cloud storage, web browsers, email)

3.6. Data Analysis~~The function, application, capabilities, and limitations of computers~~

3.6.1. Collecting, aggregating, cleaning, and modeling data

3.6.2. Using simulations, visualizations, and statistical models to perform exploratory data analysis

3.7. Algorithms and Programming~~The social and ethical implications of computers and their related technology~~

3.7.1. Fluency in at least one high-level language used in current pedagogy including variables, data types, creating and using methods, passing data between methods, control structures, and data structures
3.7.2. Programming languages, including the definition and structure of languages and comparison of existing high-level languages, particularly including object-oriented program design
3.7.3. The specification, design, implementation, testing, modification, and debugging of software
3.7.4. Apply problem-solving strategies such as design specification, top-down design, step-wise refinement, object-oriented design
3.7.5. Algorithm analysis using big-O notation to evaluate best-, average-, and worst-case space and time techniques
3.7.6. Important programming concepts such as modularity, abstraction, recursion, libraries and Application Programming Interfaces (APIs)

3.8. Impacts of Computing~~The mathematical principles which are the basis of many computer applications, including algebra, set theory, coordinate systems and graphs, matrices, and probability and statistics~~
3.8.1. Ethical acquisition (e.g., citing sources using established methods) and acceptable versus unacceptable use of information (e.g., privacy, hacking, piracy, vandalism, viruses, current laws and regulations).
3.8.2. Intellectual property rights and related issues (e.g., copyright laws, fair use, patents, trademarks) when using, manipulation, and editing electronic data.
3.8.3. Issues related to the equitable use of technology (e.g. gender, ethnicity, language, disabilities, access to technology)
3.8.4. Digital citizenship, digital footprints, and other ways technology is shaping culture and social interactions
3.8.5. Identifying and avoiding online threats including phishing schemes, sextortion, and identity theft among others.

3.9. The concepts, vocabulary, and issues found in two or more of the sub-disciplines of computer science (including but not limited to: abstract data types, advanced computer science algorithms, computer architecture, networks and data communications, physical computing, digital forensics, machine learning)~~computer architecture, artificial intelligence, data and knowledge bases, ethics, graphics, human-computer interaction, networks and data communication, programming languages, and software engineering)~~

~~The specification, design, implementation, testing, modification, and debugging of software~~

4. Performance Standards:

5. **Implements an inquiry-based computer science curriculum that integrates conceptual understanding and skill development. Specifically, the educator:**

5.1. Plans and implements instruction that allows students to use computer science in problem-solving and decision-making situations~~Designs and implements instructional activities for students that reinforce the topics, concepts, and skills central to computer science (listed above)~~
5.2. Keeps current with the use of technology in education and issues related to legal and ethical use of technology resources

~~Develops appropriate assessment criteria for student-developed software based on software quality attributes (e.g., reusability, maintainability, testability, etc.)~~

5.3. Designs and implements activities which reinforce verbal and written technical communication skills central to computer sciecne

5.4. The ability to use the basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solutions, testing, evaluation).~~Recognizes and supports creative and alternative solutions~~

5.5. Uses effective management strategies for teaching computer science (e.g. laboratory work, cooperative learning, electronic communication)

5.6. Uses appropriate instructional strategies for teaching computer science (e.g., case studies, role-playing, manipulatives, visualizations, simulations, modeling)

6. Additional Requirements:

6.1. A minor in Computer Science, or the equivalent, in undergraduate and/or graduate coursework.