Readopt with amendment Ed 507.54, effective 11-14-17, (Document #12419), to read as follows:

Ed 507.54 Computer Science Educator.

(a) A candidate seeking a credential as a computer science educator for grades K-12 shall meet the following education requirements:

(1) Either:

a. Have at least a bachelor's degree; or

b. Hold a current industry-recognized credential as defined in RSA 200-O:2, II; and

(2) Meet the requirements of (c) below.

(b) Candidates seeking a credential shall use one of the alternatives in Ed 505.05 – Ed 505.07 after meeting the requirements of (c) below.

(c) A candidate seeking a credential as a computer science educator for grades K-12 shall have the following skills, competencies, and knowledge through a combination of learning activities such as, but not limited to, academic and supervised clinical experience in the following areas:

(1) In the area of impacts of computing, the candidate shall have the ability to:

a. Use computing to:

1. Express creativity;

2. Solve problems;

3. Communicate;

4. Collaborate; and

5. Innovate in a variety of fields and careers;

b. Assess obstacles to equal access to computing as well as the impacts of these obstacles;

c. Assess impacts of computing innovations and practices with respect to:

1. Beneficial and harmful effects;

2. Ethical practices; and

3. Privacy, security, and legal issues; and

d. Keep current with knowledge on emerging technologies and their potential impacts;

(2) In the area of algorithms and computational thinking, the candidate shall have the ability to:

a. Analyze algorithmic processes and develop algorithms using:

1. Pattern recognition;

2. Problem decomposition; and

3. Abstraction;

b. Convert between binary, decimal, and hexadecimal number systems;

c. Use different fonts to represent, develop and analyze algorithms including, but not limited to natural language, flowcharts, and pseudocode;

d. Use heuristic solutions to address computing limitations including, but not limited to, time, space, and solvability;

e. Use standard algorithms including, but not limited to, searching and sorting algorithms and analyze implementations of those algorithms for correctness, efficiency, and clarity; and

f. Use simple recursive algorithms including, but not limited to, fractals, Zeno's paradox, and Towers of Hanoi;

(3) In the area of programming, the candidate shall have the ability to:

a. Write and modify computer programs in block-based and at least one text-based programming language;

b. Analyze computer programs in terms of:

1. Correctness;
2. Usability;
3. Readability;
4. Extensibility;
5. Modifiability; and
6. Reusability;

c. Program using the following elements:

1. Basic control structures;
2. Standard operators;
3. Variables and data types;
4. Functions; and
5. Data structures;

d. Write event-driven programs that respond to external events including, but not limited to, sensors, messages, and clicks;

e. Use libraries and APIs;

f. Use and evaluate multiple development and execution environments;

g. Facilitate collaboration in the development of software;

h. Program user interactions with graphical and other user interface components;

i. Demonstrate knowledge of various software development models;

j. Model practices in software development, including:

1. User requirements analysis;
2. Program design;
3. Implementation and documentation;
4. Testing and debugging; and
5. Evolution driven by feedback;

k. Develop object-oriented programs; and

l. Model the process of program compilation and interpretation;

(4) In the area of data and analysis, the candidate shall have the ability to:

a. Model concepts of processing data, including:

1. Encoding;
2. Compression; and
3. Encryption;

b. Create and use models and simulations; and

c. Work with data using computational tools, including to:

1. Collect, aggregate, and generate;
2. Store, manage, and manipulate; and
3. Process, analyze, and visualize; and

(5) In the area of computing systems and networks, the candidate shall have the ability to:

a. Evaluate and compare computing systems using various criteria;

b. Evaluate and compare local, network, and cloud computing and storage;

c. Model computer networks in terms of:

1. Protocol stack components; and
2. Network protocols, such as:

(i) Transmission control protocol/internet protocol (TCP/IP);

(ii) Domain name servers (DNS); and

(iii) Hypertext transfer protocol secure (HTTPS); and

d. Demonstrate fundamental concepts of cybersecurity including, but not limited to, confidentiality, integrity, availability, non-repudiation, and authentication.

Appendix I

Rule	**Statute**
Ed 507.54	RSA 186:11, X(a); RSA 200-O:4, I