Overton Public School

Computer Science and Literacy: Scope and Sequence

Digital Citizenship

NE K-12 Technology Scope & Sequence	✓	← *Standard may be integrated*							I = Introduce	R = Reinforce	M = Master
		K	1	2	3	4	5	6	7	8	9
Responsible Use											
Demonstrate compliance of Responsible Use Policy and classroom rules regarding technology use and networks.	✓	I	I	I	R	R	R	R	M	M	M
Explain responsible uses of technology and digital information and describe potential consequences of inappropriate use.	✓	I	I	I	R	R	R	R	M	M	M
Identify and explain the strategies for the safe and efficient use of computers (passwords, virus protection software, etc.).				I	I	R	R	R	M	M	M
Demonstrate safe email practices and appropriate email etiquette.	✓				I	I	R	R	M	M	M
Identify cyberbullying and describe strategies to deal with such a situation.	✓		I	I	I	R	R	M	M	M	M
Explore social and ethical impacts of technology.	✓	I	I	I	R	R	M	M	M	M	M
Recognize and describe the potential risks and dangers associated with online communication.	✓	I	I	I	R	R	M	M	M	M	M
Give examples of hardware and software that enable people with disabilities to use technology.						I	I	R	R	M	M
Analyze and explain how media and data can be used to distort, exaggerate, and misinterpret information.	✓				I	I	R	R	R	M	M
Explain the potential risks associated with the use of networked digital environments (Internet, cell phones, wireless networks) and sharing personal information.						I	I	R	R	M	M
Copyright											
Explain fair use guidelines for copyrighted material (images, music, videos, etc.)	✓			I	I	R	R	R	M	M	M
ISTE Standards for Students											
Students cultivate and manage their digital identity and are aware of the permanence of their actions in the digital world.				I	I	I	R	R	M	M	M
Students engage in positive, safe, legal, and ethical behavior when using technology including social interactions online or when using networked devices.	✓	I	I	I	R	R	R	R	M	M	M

Students demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.	✓			I	I	I	R	R	M	M	M
Students manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.				I	I	I	R	R	M	M	M

Computer Literacy

NE K-12 Technology Scope & Sequence		*K*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Keyboarding											
Use proper posture and ergonomics.				I	I	I	R	R	R	M	M
Locate and use letter and number keys with left and right hand placement.				I	I	I	R	R	R	M	M
Locate and use correct finger/hand for spacebar, enter, and shift key.				I	I	I	R	R	R	M	M
Gain proficiency and speed in keyboarding. (Type 5 WPM per grade level beginning at 2nd grade.)				I (5)	I (10)	I (15)	R (20)	R (25)	R (30)	M (35)	M (40)
File Management											
Organize files and folders.	✓			I	I	R	R	M	M	M	M
Manage files and save documents.	✓			I	I	R	R	M	M	M	M
Operate Basic Device Functionality											
Turn on the computer.	✓	I	I	I	R	R	M	M	M	M	M
Login and logoff the computer.	✓	I	I	I	R	R	M	M	M	M	M
Use a pointing device to click menus and icons.	✓	I	I	I	R	R	M	M	M	M	M
Open programs, web apps, and documents.	✓	I	I	I	R	R	M	M	M	M	M
Use buttons and media players.	✓	I	I	I	R	R	M	M	M	M	M
Hardware and Software											
Demonstrate an understanding of the relationship between hardware and software.		I	I	I	R	R	M	M	M	M	M
Troubleshoot basic hardware and software problems.	✓						I	I	R	R	M
Identify major computer components.		I	I	I	R	R	R	R	R	M	M
Describe the components and functions of computers and networks.					I	I	R	R	R	M	M
Apply strategies for identifying and solving routine problems that occur during everyday computer use.	✓	I	I	I	R	R	R	R	R	M	M
Word Processing											
Write, edit, save, and print documents in one sitting.	✓		I	I	R	R	M	M	M	M	M
Use menu/toolbar functions, such as font size, font style, and line spacing to format a document.	✓		I	I	R	R	M	M	M	M	M
Highlight, copy, and paste text.	✓		I	I	R	R	M	M	M	M	M

Copy, paste, insert, and resize images within the documents and from outside sources.	✓		I	I	R	R	M	M	M	M	M	
Proofread and edit writing using appropriate resources (spell checker, grammar checker, thesaurus).	✓		I	I	R	R	M	M	M	M	M	
Demonstrate the use of intermediate features in word processing applications (i.e. tabs, indents, bullets, numbers, tables, headers, footers).	✓						I	I	R	R	M	
Apply advanced formatting and page layout features when appropriate (i.e. columns, templates, styles) to improve the appearance of documents and projects.							I	I	R	R	M	
Use the comment function in review for peer editing.	✓							I	R	M	M	
Use the track changes feature in review for peer editing of documents.	✓							I	R	M	M	
Spreadsheets												
Enter and edit data and perform calculations using formulas.	✓						I	I	R	R	M	
Demonstrate an understanding of recording, organizing, and graphing information.	✓			I	I	R	R	R	M	M	M	
Identify and explain terms and concepts related to spreadsheets (i.e. cells, columns, rows, values, charts, graphs).	✓			I	I	R	R	R	M	M	M	
Use mathematical symbols appropriately.	✓		I	I	R	R	M	M	M	M	M	
Use spreadsheets to make predictions, solve problems, and draw conclusions.							I	I	R	R	M	
Use spreadsheets to calculate, graph, organize, and present data in a variety of real world settings.	✓						I	I	R	R	M	
Enter formulas and functions in spreadsheet applications.	✓						I	I	R	R	M	
Use and modify spreadsheets to analyze data and propose solutions.							I	I	R	R	M	
Use the functions and tools of a spreadsheet application (i.e. autofill, sort, filter, find).							I	I	R	R	M	
Presentation Tools												
Create, edit, and format text.	✓				I	I	R	R	M	M	M	
Create a series of slides and organize them to present research or convey data.	✓				I	I	R	R	M	M	M	
Copy, paste, insert, and resize images within the slides and from outside sources.	✓				I	I	R	R	M	M	M	
Create presentations for a variety of audiences and purposes with the use of appropriate transitions and animations to add interest.	✓						I	I	R	R	M	

Digital Media											
Watch videos and use play, pause, rewind, and forward buttons.	✓	I	I	R	R	M	M	M	M	M	M
Watch videos and use play, pause, rewind, and forward buttons while taking notes.	✓					I	I	R	R	M	M
Use painting/drawing tools and other applications to create and edit work.	✓	I	I	I	R	R	R	M	M	M	M
Create media for a variety of audiences and purposes with the use of appropriate transitions and animations to add interest.	✓					I	I	R	R	M	M
Independently use appropriate technology tools (graphic organizers, audio, and video) to define problems and propose hypotheses.	✓						I	I	R	R	M
Research											
Use Internet browsers, search engines, and online directories, compare the differences, and explain how they disseminate information.		I	I	I	R	R	M	M	M	M	M
Identify careers and industry opportunities.	✓		I	I	I	I	R	R	M	M	M
Perform basic searches on a database (i.e. library card catalogue) to locate information.	✓	I	I	I	R	R	M	M	M	M	M
Use content-specific technology tools to gather and analyze data.	✓		I	I	I	I	R	R	M	M	M
Identify and analyze the purpose of a media message (inform, persuade, entertain).	✓	I	I	I	R	R	M	M	M	M	M
Identify and explain current hardware and software trends.					I	I	R	R	R	M	M
Use Internet browsers, search engines, and online directories, compare the differences, and explain how they rank results.									I	R	M
Write correct in-text citations and reference lists for text and images gathered from electronic sources.	✓								I	R	M
Use Internet browsers to access information (i.e. enter a URL, access links, create bookmarks, print webpages).	✓	I	I	I	R	R	M	M	M	M	M
Communications and Collaboration											
Collaborate using technology.	✓			I	R	M	M	M	M	M	M
Use a variety of age-appropriate technologies to communicate and exchange ideas.	✓		I	I	R	M	M	M	M	M	M
Create projects that use text, graphics, audio, and video to communicate ideas.	✓			I	R	M	M	M	M	M	M
Evaluate presentations for organization, content, design, and appropriateness of citation.	✓			I	I	R	R	R	M	M	M
Plan and implement a collaborative project with other students using technology tools (i.e. email, discussion forums, video conference).	✓							I	R	M	M

Computer Science

NE K-12 Technology Scope & Sequence

Computational Thinking		K	1	2	3	4	5	6	7	8	9
Create algorithms, or series of ordered steps, to solve problems.	✓	I	I	I	R	R	R	M	M	M	M
Decompose a problem into smaller more manageable parts.			I	I	R	R	M	M	M	M	M
Collect, analyze, and represent data effectively.	✓			I	I	R	R	R	M	M	M
Demonstrate an understanding of how information is represented, stored, and processed by a computer.					I	I	I	R	R	M	M
Optimize an algorithm for execution by a computer.							I	I	R	R	M
Create simulations/models to understand natural phenomena and test hypotheses.	✓							I	R	R	M
Evaluate algorithms by their efficiency, correctness, and clarity.	✓								I	R	M

CSTA K-12 Computer Science Standards (identifier: subconcept/core practice)

	Level 1A			Level 1B			Level 2			Level 3A
Computing Systems	K (I)	1 (R)	2 (M)	3 (I)	4 (R)	5 (M)	6 (I)	7 (R)	8 (M)	9 (I)
1A-CS-01: Devices/1.1	Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.									
1A-CS-02: Hardware & Software/7.2	Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).									
1A-CS-03: Troubleshooting/6.2, 7.2	Describe basic hardware and software problems using accurate terminology.									
1B-CS-01: Devices/7.2				Describe how internal and external parts of computing devices function to form a system.						
1B-CS-02: Hardware & Software/4.4				Model how computer hardware and software work together as a system to accomplish tasks.						
1B-CS-03: Troubleshooting/6.2				Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.						
2-CS-01: Devices/3.3							Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.			
2-CS-02: Hardware & Software/5.1							Design projects that combine hardware and software components to collect and exchange data.			
2-CS-03: Troubleshooting/6.2							Systematically identify and fix problems with computing devices and their components.			
3A-CS-01: Devices/4.1							Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects →			

3A-CS-02: Hardware & Software/4.1				Compare levels of abstraction and interactions between application software, system software, and hardware layers.	→
3A-CS-03: Troubleshooting/6.2				Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	→
Networks and the Internet					
1A-NI-04: Cybersecurity/7.3		Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access.			
1B-NI-04: Network Communication & Organization/4.4			Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.		
1B-NI-05: Cybersecurity/3.1			Discuss real-world cybersecurity problems and how personal information can be protected.		
2-NI-04: Network Communication & Organization/4.4				Model the role of protocols in transmitting data across networks and the Internet.	
2-NI-05: Cybersecurity/7.2				Explain how physical and digital security measures protect electronic information.	
2-NI-06: Cybersecurity/4.4				Apply multiple methods of encryption to model the secure transmission of information.	
3A-NI-04: Network Communication & Organization/4.1				Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	→
3A-NI-05: Network Communication & Organization/7.2				Give examples to illustrate how sensitive data can be affected by malware and other attacks.	→
3A-NI-06: Cybersecurity/3.3				Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	→
3A-NI-07: Network Communication & Organization/6.3				Compare various security measures, considering tradeoffs between the usability and security of a computing system.	→
3A-NI-08: Cybersecurity/7.2				Explain tradeoffs when selecting and implementing cybersecurity recommendations.	→
Data and Analysis					
1A-DA-05: Storage/4.2	✓	Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.			
1A-DA-06: Collection, Visualization & Transformation/7.1, 4.4	✓	Collect and present the same data in various visual formats.			
1A-DA-07: Inference & Models/4.1	✓	Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.			
1B-DA-06: Collection, Visualization & Transformation/7.1	✓		Organize and present collected data visually to highlight relationships and support a claim.		
1B-DA-07: Inference & Models/7.1	✓		Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.		
2-DA-07: Storage/4				Represent data using multiple encoding schemes.	

2-DA-08: Collection, Visualization & Transformation/6.3			Collect data using computational tools and transform the data to make it more useful and reliable.	
2-DA-09: Inference & Models/5.3, 4.4			Refine computational models based on the data they have generated.	
3A-DA-09: Storage/4.1			Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.	→
3A-DA-10: Storage/3.3			Evaluate the tradeoffs in how data elements are organized and where data is stored.	→
3A-DA-11: Collection, Visualization & Transformation/4.4			Create interactive data visualizations using software tools to help others better understand real-world phenomena.	→
3A-DA-12: Inference & Models/4.4			Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.	→
Algorithms and Programming				
1A-AP-08: Algorithms/4.4	✓	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.		
1A-AP-09: Variables/4.4		Model the way programs store and manipulate data by using numbers or other symbols to represent information.		
1A-AP-10: Control/5.2		Develop programs with sequences and simple loops, to express ideas or address a problem.		
1A-AP-11: Modularity/3.2	✓	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.		
1A-AP-12: Program Development/5.1, 7.2		Develop plans that describe a program's sequence of events, goals, and expected outcomes.		
1A-AP-13: Program Development/7.3		Give attribution when using the ideas and creations of others while developing programs.		
1A-AP-14: Program Development/6.2		Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.		
1A-AP-15: Program Development/7.2		Using correct terminology, describe steps taken and choices made during the iterative process of program development.		
1B-AP-08: Algorithms/6.3, 3.3	✓		Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	
1B-AP-09: Variables/5.2			Create programs that use variables to store and modify data.	
1B-AP-10: Control/5.2			Create programs that include sequences, events, loops, and conditionals.	
1B-AP-11: Modularity/3.2			Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	
1B-AP-12: Modularity/5.3			Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	
1B-AP-13: Program Development/1.1, 5.1			Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	

1B-AP-14: Program Development/5.2, 7.3	Observe intellectual property rights and give appropriate attribution when creating or remixing programs.			
1B-AP-15: Program Development/6.1, 6.2	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.			
1B-AP-16: Program Development/2.2	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.			
1B-AP-17: Program Development/7.2	Describe choices made during program development using code comments, presentations, and demonstrations.			
2-AP-10: Algorithms/4.4, 4.1		Use flowcharts and/or pseudocode to address complex problems as algorithms.		
2-AP-11: Variables/5.1, 5.2		Create clearly named variables that represent different data types and perform operations on their values.		
2-AP-12: Control/5.1, 5.2		Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.		
2-AP-13: Modularity/3.2		Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.		
2-AP-14: Modularity/4.1, 4.3		Create procedures with parameters to organize code and make it easier to reuse.		
2-AP-15: Program Development/2.3, 1.1		Seek and incorporate feedback from team members and users to refine a solution that meets user needs.		
2-AP-16: Program Development/4.2, 5.2, 7.3		Incorporate existing code, media, and libraries into original programs, and give attribution.		
2-AP-17: Program Development/6.1		Systematically test and refine programs using a range of test cases.		
2-AP-18: Program Development/2.2		Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.		
2-AP-19: Program Development/7.2		Document programs in order to make them easier to follow, test, and debug.		
3A-AP-13: Algorithms/5.2			Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	→
3A-AP-14: Variables/4.1			Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	→
3A-AP-15: Control/5.2			Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.	→
3A-AP-16: Control/5.2			Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to iniate instructions.	→
3A-AP-17: Control/3.2			Decompose problems into smaller components through systemic analysis, using constructs such as procedures, modules, and/or objects.	→

3A-AP-18: Modularity/5.2				Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	→
3A-AP-19: Modularity/5.1				Systematically design and develop programs for broad audiences by incorporating feedback from users.	→
3A-AP-20: Program Development/7.3				Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	→
3A-AP-21: Program Development/6.3				Evaluate and refine computaional artifacts to make them more usable and accessible.	→
3A-AP-22: Program Development/2.4				Design and develop computational artifacts working in team roles using collaborative tools.	→
3A-AP-23: Program Development/7.2				Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	→
Impacts of Computing					
1A-IC-16: Culture/7	✓	Compare how people live and work before and after the implementation or adoption of new computing technology.			
1A-IC-17: Social Interactions/2.1	✓	Work respectfully and responsibly with others online.			
1A-IC-18: Safety, Law & Ethics/7.3	✓	Keep login information private, and log off of devices appropriately.			
1B-IC-19: Culture/3.1	✓		Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.		
1B-IC-19: Culture/1.2			Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.		
1B-IC-20: Social Interactions/1.1			Seek diverse perspectives for the purpose of improving computational artifacts.		
1B-IC-21: Safety, Law & Ethics/7.3	✓		Use public domain or creative commons media, and refrain from copying or using material created by others without permission.		
2-IC-20: Culture/7.2				Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	
2-IC-21: Culture/1.2				Discuss issues of bias and accessibility in the design of existing technologies.	
2-IC-22: Social Interactions/2.4, 5.2				Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	
2-IC-23: Safety, Law & Ethics/7.2	✓			Describe tradeoffs between allowing information to be public and keeping information private and secure.	
3A-IC-24: Culture/1.2					Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. →
3A-IC-25: Culture/1.2					Test and refine computational artifacts to reduce bias and equity deficits. →
3A-IC-26: Culture/3.1					Demonstrate ways a given algorithm applies to problems across disciplines. →

3A-IC-27: Social Interactions/2.4			Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.	→
3A-IC-28: Safety, Law & Ethics/7.3	✓		Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	→
3A-IC-29: Safety, Law & Ethics/7.2			Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	→
3A-IC-30: Safety, Law & Ethics/7.3			Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	→

References

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. Journal of Educational Technology & Society, 19(3), 47-57.

Blomstedt, M. L. (2018, December 17). Computer science education in Nebraska. Retrieved from Nebraska Department of Education: cdn.education.ne.gov/wp-content/uploads/2019/05/Computer-Science-Letter.pdf

Buckler, C., Koperski, K., & Loveland, T. R. (2018, December/January). Is computer science compatible with technological literacy? Technology and Engineering Teacher, pp. 15-20.

Cortina, T. J. (2015, March). Broadening participation: Reaching a broader population of students through "unplugged" activities. Communications of the ACM, 58(3), 25-27.

CSTA. (2017). K-12 Computer Science Standards. Retrieved from Computer Science Teachers Association: www.csteachers.org/page/standards

CyberWise. (2020). Our top digital citizenship resources. Retrieved from CyberWise: http://cyberwise.org/digital-citizenship-resources

Domain. (2020). Retrieved from Oxford Online Dictionary: https://en.oxforddictionaries.com/definition/domain

Fluck, A., Webb, M., Cox, M., Charoula, A., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for computer science in the school curriculum. Journal of Educational Technology & Society, 19(3), 38-46.

Gandal, M. (1995). Why we need academic standards. Educational Leadership, 53(1), pp. 84-86.

Guzdial, M. (2019, August). Block-based programming in computer science education. Communications of the ACM, 62(8), 22-25.

Hazzan, O., Lapidot, T., & Ragonis, N. (2014). Guide to teaching computer science: An activity-based approach. London: Springer-Verlag.

Heinlein, L. M., & Shinn, M. (2000). School mobility and student achievement in an urban setting. Psychology in the Schools, 37(4), 349-357.

Immigration Policy Center. (2013). Always in demand: The economic contributions of immigrant scientists and engineers. Washington, D.C.: American Immigration Council.

ISTE. (2016). ISTE standards for students. Retrieved from International Society for Technology in Education: http://iste.org/standards/for-students

K-12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org

Kindsiko, E., Aidla, A., Poltimäe, H., & Türk, K. (2020). They only teach us word and excel! Trames, 24(1), 53-69.

Krauss, J., & Prottsman, K. (2017). Computational thinking and coding for every student. Thousand Oaks, CA: Corwin.

NCWIT. (2016). Engagement practices framework. Retrieved from National Center for Women & Information Technology: http://ncwit.org/engagement-practices-framework

NDE. (2018). Nebraska K-12 Technology Scope & Sequence. Retrieved from Nebraska Department of Education: www.education.ne.gov/nce/cis

Ribble, M. (2015). Digital citizenship in schools: Nine elements all students should know. Eugene, OR: International Society for Technology in Education.

Simons, P. R. (1999). Transfer of learning: Paradoxes for learners. International Journal of Educational Research, 31(7), 577-589.

Wilson, C. (2013, November). Making computer science count. Communications of the ACM, 56(11), 32-33.