June 2023

Skill Struck®

Skill Struck's alignment to

Michigan K-12 Standards Computer Science

Legend

✅ = Standard aligned

🔹 = Not currently aligned

Standard	Status
1A-CS-01 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. Subconcept: Devices; Practice 1.1	✅
1A-CS-02 Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). Subconcept: Hardware & Software; Practice 7.2	✅
1A-CS-03 Describe basic hardware and software problems using accurate terminology. Subconcept: Troubleshooting; Practice 6.2, 7.2	✅

1A-NI-04 Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. Subconcept: Cybersecurity; Practice 7.3	✅
1A-DA-05 Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. Subconcept: Storage; Practice 4.2	✅
1A-DA-06 Collect and present the same data in various visual formats. Subconcept: Collection, Visualization & Transformation; Practice 7.1, 4.4	✅
1A-DA-07 Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. Subconcept: Inference & Models; Practice 4.1	✅
1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. Subconcept: Algorithms; Practice 4.4	✅
1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information. Subconcept: Variables; Practice 4.4	✅
1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. Subconcept: Control; Practice 5.2	✅
1A-AP-11	✅

Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. Subconcept: Modularity; Practice 3.2	
1A-AP-12 Develop plans that describe a programís sequence of events, goals, and expected outcomes. Subconcept: Program Development; Practice 5.1, 7.2	✅
1A-AP-13 Give attribution when using the ideas and creations of others while developing programs Subconcept: Program Development; Practice 7.3	✅
1A-AP-14 Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. Subconcept: Program Development; Practice 6.2	✅
1A-AP-15 Using correct terminology, describe steps taken and choices made during the iterative process of program development. Subconcept: Program Development; Practice 7.2	✅
1A-IC-16 Compare how people live and work before and after the implementation or adoption of new computing technology. Subconcept: Culture; Practice 7	✅
1A-IC-17 Work respectfully and responsibly with others online. Subconcept: Social Interactions; Practice 2.1	✅
1A-IC-18 Keep login information private, and log off of devices appropriately.	✅

Subconcept: Safety Law & Ethics; Practice 7.3	
1B-CS-01 Describe how internal and external parts of computing devices function to form a system. Subconcept: Devices; Practice 7.2	✅
1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks. Subconcept: Hardware & Software; Practice 4.4	✅
1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. Subconcept: Troubleshooting; Practice 6.2	✅
1B-NI-04 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. Subconcept: Network Communication & Organization; Practice 4.4	✅
1B-NI-05 Discuss real-world cybersecurity problems and how personal information can be protected. Subconcept: Cybersecurity; Practice 3.1	✅
1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim. Subconcept: Collection, Visualization & Transformation; Practice 7.1	✅
1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict	✅

outcomes, or communicate an idea. Subconcept: Inference & Models; Practice 7.1	
1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. Subconcept: Algorithms; Practice 6.3, 3.3	✅
1B-AP-09 Create programs that use variables to store and modify data. Subconcept: Variables; Practice 5.2	✅
1B-AP-10 Create programs that include sequences, events, loops, and conditionals. Subconcept: Control; Practice 5.2	✅
1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. Subconcept: Modularity; Practice 3.2	✅
1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. Subconcept: Modularity; Practice 5.3	✅
1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. Subconcept: Program Development; Practice 1.1, 5.1	✅
1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. Subconcept: Program Development; Practice 5.2, 7.3	✅

1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as Intended. Subconcept: Program Development; Practice 6.1, 6.2	✅
1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. Subconcept: Program Development; Practice 2.2	✅
1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations. Subconcept: Program Development; Practice 7.2	✅
1B-IC-18 Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices. Subconcept: Culture; Practice 3.1	✅
1B-IC-19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. Subconcept: Culture; Practice 1.2	✅
1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts Subconcept: Social Interactions; Practice 1.1	✅
1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.	✅

Subconcept: Safety Law & Ethics; Practice 7.3	
2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. Subconcept: Devices; Practice 3.3	✅
2-CS-02 Design projects that combine hardware and software components to collect and exchange data. Subconcept: Hardware & Software; Practice 5.1	✅
2-CS-03 Systematically identify and fix problems with computing devices and their components. Subconcept: Troubleshooting; Practice 6.2	✅
2-NI-04 Model the role of protocols in transmitting data across networks and the Internet. Subconcept: Network Communication & Organization; Practice 4.4	✅
2-NI-05 Explain how physical and digital security measures protect electronic information. Subconcept: Cybersecurity; Practice 7.2	✅
2-NI-06 Apply multiple methods of encryption to model the secure transmission of information. Subconcept: Cybersecurity; Practice 4.4	✅
2-DA-07 Represent data using multiple encoding schemes. Subconcept: Storage; Practice 4	✅

2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable. Subconcept: Collection, Visualization & Transformation; Practice 6.3	✅
2-DA-09 Refine computational models based on the data they have generated. Subconcept: Inference & Models; Practice 5.3, 4.4	✅
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. Subconcept: Algorithms; Practice 4.4, 4.1	✅
2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. Subconcept: Variables; Practice 5.1, 5.2	✅
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. Subconcept: Control; Practice 5.1, 5.2	✅
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. Subconcept: Modularity; Practice 3.2	✅
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. Subconcept: Modularity; Practice 4.1, 4.3	✅
2-AP-15	✅

Seek and incorporate feedback from team members and users to refine a solution that meets user needs. Subconcept: Program Development; Practice 2.3, 1.1	
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution. Subconcept: Program Development; Practice 4.2, 5.2, 7.3	✅
2-AP-17 Systematically test and refine programs using a range of test cases. Subconcept: Program Development; Practice 6.1	✅
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. Subconcept: Program Development; Practice 2.2	✅
2-AP-19 Document programs in order to make them easier to follow, test, and debug. Subconcept: Program Development; Practice 7.2	✅
2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. Subconcept: Culture; Practice 7.2	✅
2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies. Subconcept: Culture; Practice 1.2	✅
2-IC-22 Collaborate with many contributors through strategies such as	✅

crowdsourcing or surveys when creating a computational artifact. Subconcept: Social Interactions; Practice 2.4, 5.2	
2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure. Subconcept: Safety Law & Ethics; Practice 7.2	✅
3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. Subconcept: Devices; Practice 4.1	✅
3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers. Subconcept: Hardware & Software; Practice 4.1	✅
3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. Subconcept: Troubleshooting; Practice 6.2	✅
3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. Subconcept: Network Communication & Organization; Practice 4.1	✅
3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks. Subconcept: Network Communication & Organization; Practice 7.2	✅
3A-NI-06 Recommend security measures to address various scenarios based on	✅

factors such as efficiency, feasibility, and ethical impacts. Subconcept: Cybersecurity; Practice 3.3	
3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. Subconcept: Network Communication & Organization; Practice 6.3	✅
3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations. Subconcept: Cybersecurity; Practice 7.2	✅
3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. Subconcept: Storage; Practice 4.1	✅
3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored. Subconcept: Storage; Practice 3.3	✅
3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena. Subconcept: Collection, Visualization & Transformation; Practice 4.4	✅
3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. Subconcept: Inference & Models; Practice 4.4	✅
3A-AP-13 Create prototypes that use algorithms to solve computational problems	✅

by leveraging prior student knowledge and personal interests. Subconcept: Algorithms; Practice 5.2	
3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. Subconcept: Variables; Practice 4.1	✅
3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. Subconcept: Control; Practice 5.2	✅
3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. Subconcept: Control; Practice 5.2	✅
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. Subconcept: Control; Practice 3.2	✅
3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. Subconcept: Modularity; Practice 5.2	✅
3A-AP-19 Systematically design and develop programs for broad audiences by incorporating feedback from users. Subconcept: Modularity; Practice 5.1	✅

3A-AP-20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. Subconcept: Program Development; Practice 7.3	✅
3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible. Subconcept: Program Development; Practice 6.3	✅
3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools. Subconcept: Program Development; Practice 2.4	✅
3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. Subconcept: Program Development; Practice 7.2	✅
3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. Subconcept: Culture; Practice 1.2	✅
3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits. Subconcept: Culture; Practice 1.2	✅
3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines. Subconcept: Culture; Practice 3.1	✅
3A-IC-27	✅

Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. Subconcept: Social Interactions; Practice 2.4	
3A-IC-28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation. Subconcept: Safety Law & Ethics; Practice 7.3	✅
3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. Subconcept: Safety Law & Ethics; Practice 7.2	✅
3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. Subconcept: Safety Law & Ethics; Practice 7.3	✅
3B-CS-01 Categorize the roles of operating system software. Subconcept: Hardware & Software; Practice 7.2	✅
3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components. Subconcept: Troubleshooting; Practice 7.2	✅
3B-NI-03 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology). Subconcept: Network Communication & Organization 7.2	✅
3B-NI-04 Compare ways software developers protect devices and information from	✅

unauthorized access. Subconcept: Cybersecurity; Practice 7.2	
3B-DA-05 Use data analysis tools and techniques to identify patterns in data representing complex systems. Subconcept: Collection Visualization & Transformation; Practice 4.1	✅
3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information. Subconcept: Collection Visualization & Transformation; Practice 7.2	✅
3B-DA-07 Evaluate the ability of models and simulations to test and support the refinement of hypotheses. Subconcept: Inference & Models; Practice 4.4	✅
3B-AP-08 Describe how artificial intelligence drives many software and physical systems. Subconcept: Algorithms; Practice 7.2	✅
3B-AP-09 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem. Subconcept: Algorithms; Practice 5.3	✅
3B-AP-10 Use and adapt classic algorithms to solve computational problems. Subconcept: Algorithms; Practice 4.2	✅
3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity. Subconcept: Algorithms; Practice 4.2	✅

3B-AP-12 Compare and contrast fundamental data structures and their uses. Subconcept: Variables; Practice 4.2	✅
3B-AP-13 Illustrate the flow of execution of a recursive algorithm. Subconcept: Control; Practice 3.2	✅
3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects. Subconcept: Modularity; Practice 5.2	✅
3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. Subconcept: Modularity; Practice 4.1	✅
3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs. Subconcept: Modularity; Practice 5.3	✅
3B-AP-17 Plan and develop programs for broad audiences using a software life cycle process. Subconcept: Program Development; Practice 5.1	✅
3B-AP-18 Explain security issues that might lead to compromised computer programs. Subconcept: Program Development; Practice 7.2	✅
3B-AP-19 Develop programs for multiple computing platforms.	✅

Subconcept: Program Development; Practice 5.2	
3B-AP-20 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. Subconcept: Program Development; Practice 2.4	✅
3B-AP-21 Develop and use a series of test cases to verify that a program performs according to its design specifications. Subconcept: Program Development; Practice 6.1	✅
3B-AP-22 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). Subconcept: Program Development; Practice 5.3	✅
3B-AP-23 Evaluate key qualities of a program through a process such as a code review. Subconcept: Program Development; Practice 6.3	✅
3B-AP-24 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. Subconcept: Program Development; Practice 7.2	✅
3B-IC-25 Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society. Subconcept: Culture; Practice 6.1, 1.2	✅
3B-IC-26 Evaluate the impact of equity, access, and influence on the distribution of	✅

computing resources in a global society. Subconcept: Culture; Practice 1.2	
3B-IC-27 Predict how computational innovations that have revolutionized aspects of our culture might evolve. Subconcept: Culture; Practice 7.2	✅
3B-IC-28 Debate laws and regulations that impact the development and use of software. Subconcept: Safety Law & Ethics; Practice 3.3, 7.3	✅