Mississippi Computer Science

Grades 3-5

Computing Systems CS.1B

CS Computing Systems CS.1B

Conceptual understanding: People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended.

1 **Describe how internal and external parts of computing devices function to form a system. [DEVICES] (P7.2) Computing devices often depend on other devices or components. For example, a robot depends on a physically attached light sensor to detect changes in brightness, whereas the light sensor depends on the robot for power. Keyboard input or a mouse click could cause an action to happen or information to be displayed on a screen; this could only happen because the computer has a processor to evaluate what is happening externally and produce corresponding responses.** CS.1B.1

- a Students should describe how devices and components interact using correct terminology. CS.1B.1A

2 **Model how computer hardware and software work together as a system to accomplish tasks. [HARDWARE & SOFTWARE] (P4.4) In order for a person to accomplish tasks with a computer, both hardware and software are needed. At this stage, a model should only include the basic elements of a computer system, such as input, output, processor, sensors, and storage.** CS.1B.2

- a Students should model how computer hardware and software work together to accomplish tasks. Students could draw a model on paper or in a drawing program, program an animation to demonstrate it, or demonstrate it by acting this out in some way. CS.1B.2A

3 **Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. [TROUBLESHOOTING] (P6.2) Although computing systems may vary, common troubleshooting strategies can be used on all of them.** CS.1B.3

- a Students should be able to identify common hardware and software problems. Types of problems students might encounter include the device not responding, no power, no network, app crashing, no sound, or password entry not working. CS.1B.3A
- b Students should identify and implement various troubleshooting strategies. Such strategies may include rebooting the device, checking for power, checking network availability, closing and reopening an app, making sure speakers are turned on or headphones are plugged in, and making sure that the caps lock key is not on, to solve these problems, when possible. CS.1B.3B

Networks and the Internet NI.1B

NI Networks and the Internet NI.1B

Conceptual understanding: Computing devices typically do not operate in isolation. Networks connect computing devices to share information and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication and facilitating innovation.

1 **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. [NETWORK COMMUNICATION & ORGANIZATION] (P4.4) Information is sent and received over physical or wireless paths. It is broken down into smaller pieces called packets, which are sent independently and reassembled at the destination.** NI.1B.1

- a Students should demonstrate their understanding of how information flows over networks and the Internet. This could be accomplished, for instance, by drawing a model of the way packets are transmitted, programming an animation to show how packets are transmitted, or demonstrating this through an unplugged activity which has them act it out in some way. NI.1B.1A

2 **Discuss real-world cybersecurity problems and how personal information can be protected. [CYBERSECURITY] (P3.1, 7.3) Just as we protect our personal property offline, we also need to protect our devices and the information stored on them. Information can be protected using various security measures. These measures can be physical and/or digital.** NI.1B.2

- a Students should be able to explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. Learning to protect one's device or information from unwanted use by others is an essential first step in learning about cybersecurity. Students are not required to use multiple strong passwords. They should appropriately use and protect the passwords they are required to use. NI.1B.2A
- b Students should be able to list several real-world cybersecurity issues and discuss how personal information can be protected. Students could discuss or use a journaling or blogging activity to explain, orally or in writing, topics that relate to personal cybersecurity issues. Discussion topics could be based on current events related to cybersecurity or topics that are applicable to students, such as the necessity of backing up data to guard against loss, how to create strong passwords and the importance of not sharing passwords, or why we should install and keep antivirus software updated to protect data and systems. NI.1B.2B

Data and Analysis DA.1B

DA Data and Analysis DA.1B

Conceptual understanding: Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.

1 **Organize and present collected data visually to highlight relationships and support a claim. [COLLECTION, VISUALIZATION, & TRANSFORMATION] (P7.1) Raw data has little meaning on its own. Data is often sorted or grouped to provide additional clarity. Organizing data can make interpreting and communicating it to others easier. Data points can be clustered by a number of commonalities. The same data could be manipulated in different ways to emphasize particular aspects or parts of the data set. For example, a data set of sports teams could be sorted by wins, points scored, or points allowed, and a data set of weather information could be sorted by high temperatures, low temperatures, or precipitation.** DA.1B.1

- a Students should be able to collect data and present the information in an organized way to highlight relationships and support a claim. DA.1B.1A

2 **Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. [INFERENCE & MODELS] (P7.1) The accuracy of data analysis is related to how realistically data is represented. Inferences or predictions based on data are less likely to be accurate if the data is not sufficient or if the data is incorrect in some way.** DA.1B.2

- a Students should be able to refer to data to highlight or propose cause-and-effect relationships and predict outcomes when communicating an idea. For example, in order to explore the relationship between speed, time, and distance, students could operate a robot at uniform speed and at increasing time intervals to predict how far the robot travels at that speed. In order to make an accurate prediction, one or two attempts of differing times would not be enough. The robot may also collect temperature data from a sensor, but that data would not be relevant for the task. Students must also make accurate measurements of the distance the robot travels in order to develop a valid prediction. Students could record the temperature at noon each day as a basis to show that temperatures are higher in certain months of the year. If temperatures are not recorded on non-school days or are recorded incorrectly or at different times of the day, the data would be incomplete and the ideas being communicated could be inaccurate. Students may also record the day of the week on which the data was collected, but this would have no relevance to whether temperatures are higher or lower. In order to have sufficient and accurate data on which to communicate the idea, students might want to use data provided by a governmental weather agency. DA.1B.2A

3 **Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. [STORAGE] (P4.2) All information stored and processed by a computing device is referred to as data. Data can be images, text documents, audio files, software programs or apps, video files, etc.** DA.1B.3

- a Students should be able to manipulate data through their use of software to complete tasks on a computing device. For example, saving, retrieving, and deleting files are all instances of manipulating data. DA.1B.3A

Algorithms and Programming AP.1B

AP Algorithms and Programming AP.1B

Conceptual Understanding: An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. The development process to create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions.

1 **Compare and refine multiple algorithms for the same task and determine which is the most appropriate. [ALGORITHMS] (P6.3, P3.3) Different algorithms can achieve the same result, though sometimes one algorithm might be most appropriate for a specific situation.** AP.1B.1

a Students should be able to look at different ways to solve the same task and decide which would be the best solution. For example, students could use a map and plan multiple algorithms to get from one point to another. They could look at routes suggested by mapping software and change the route to something that would be better based on which route is shortest or fastest or would avoid a problem. Students might compare algorithms that describe how to get ready for school. Another example might be to write different algorithms to draw a regular polygon and determine which algorithm would be the easiest to modify or repurpose to draw a different polygon. AP.1B.1A

2 **Create programs that use variables to store and modify data. [VARIABLES] (P5.2) Variables are used to store and modify data.** AP.1B.2

a Students should understand how to use variables to store and modify data. For example, students may use mathematical operations to add to the score of a game or subtract from the number of lives available in a game. The use of a variable as a countdown timer is another example. AP.1B.2A

3 **Create programs that include sequences, events, loops, and conditionals. [CONTROL] (P5.2) Control structures specify the order (sequence) in which instructions are executed within a program and can be combined to support the creation of more complex programs. Events allow portions of a program to run based on a specific action.**

a Students should be able to create programs that include sequences, events, loops, and conditionals. For example, students could write a program to explain the water cycle. When a specific component is clicked (event), the program would show information about that part of the water cycle. Conditionals allow for the execution of a portion of code in a program when a certain condition is true. For example, students could write a math game that asks multiplication fact questions and then uses a conditional to check whether or not the answer that was entered is correct. Loops allow for the repetition of a sequence of code multiple times. For example, in a program that produces an animation about a famous historical character, students could use a loop to have the character walk across the screen as they introduce themselves. AP.1B.3A

4 **Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. [MODULARITY] (P3.2) Decomposition is the act of breaking down tasks into simpler tasks.**

a Students should be able to break down problems into smaller, simpler tasks. For example, students could create an animation by separating a story into different scenes. For each scene, they would select a background, place characters, and program actions. AP.1B.4A

5 **Modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features. [MODULARITY] (P5.3) Programs can be broken down into smaller parts, which can be incorporated into new or existing programs.** AP.1B.5

a Students should be able to modify and/or reuse portions of an existing program into their own work to create something new. For example, students could modify prewritten code from a single-player game to create a two-player game with slightly different rules, remix and add another scene to an animated story, use code to make a ball bounce from another program in a new basketball game, or modify an image created by another student. AP.1B.5A

6 **Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. [PROGRAM DEVELOPMENT] (P1.1, P5.1) Planning is an important part of the iterative process of program development.** AP.1B.6

a Students outline key features, time and resource constraints, and user expectations. AP.1B.6A

b Students should document the plan as, for example, a storyboard, flowchart, pseudocode, or story map. AP.1B.6B

7 **Observe intellectual property rights and give appropriate attribution when creating or remixing programs. [PROGRAM DEVELOPMENT] (P7.3) Intellectual property rights can vary by country, but copyright laws give the creator of a work a set of rights that prevents others from copying the work and using it in ways that they may not like.** AP.1B.7

a Students should identify instances of remixing, when ideas are borrowed and iterated upon, and credit the original creator. AP.1B.7A

b Students should also consider common licenses that place limitations or restrictions on the use of computational artifacts, such as images and music downloaded from the Internet. At this stage, attribution should be written in the format required by the teacher and should always be included on any programs shared online. AP.1B.7B

8 **Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. [PROGRAM DEVELOPMENT] (P6.1, P6.2) As students develop programs, they should continuously test those programs to see that they do what was expected and fix (debug), any errors.** AP.1B.8

a Students should be able to identify and debug simple errors in programs they create and in programs created by others. AP.1B.8A

9 **Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. [PROGRAM DEVELOPMENT] (P2.2) Collaborative computing is the process of performing a computational task by working in pairs or on teams. Because it involves asking for the contributions and feedback of others, effective collaboration can lead to better outcomes than working independently.** AP.1B.9 T

a Students should take turns in different roles during program development, such as note taker, facilitator, program tester, or “driver” of the computer. AP.1B.9A

10 **Describe choices made during program development using code comments, presentations, and demonstrations. [PROGRAM DEVELOPMENT] (P7.2) People communicate about their code to help others understand and use their programs. Another purpose of communicating one's design choices is to show an understanding of one's work.** AP.1B.10

a Students should explain code choices using comments within the code, presentations, and demonstrations. These explanations could manifest themselves as in-line code comments for collaborators and assessors or as part of a summative presentation, such as a code walk-through or coding journal. AP.1B.10A

Impacts of Computing IC.1B

IC Impacts of Computing IC.1B

Conceptual Understanding: Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions, and in turn, computing influences new cultural practices. An informed and responsible person should understand the social implications of the digital world, including equity and access to computing.

1 **Discuss computing technologies that have changed the world and express how those technologies influence and are influenced by cultural practices. [CULTURE] (P7.1) New computing technology is created and existing technologies are modified for many reasons, including to increase their benefits, decrease their risks, and meet societal needs.** IC.1B.1

a Students, with guidance from their teacher, should discuss topics that relate to the history of technology and the changes in the world due to technology. Topics could be based on current news content, such as robotics, wireless Internet, mobile computing devices, GPS systems, wearable computing, or how social media has influenced social and political changes. IC.1B.1A

2 **Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. [CULTURE] (P1.2) The development and modification of computing technology are driven by people's needs and wants and can affect groups differently. Anticipating the needs and wants of diverse end users requires students to purposefully consider potential perspectives of users with different backgrounds, ability levels, points of view, and disabilities.** IC.1B.2

a Students will demonstrate an understanding of diversity in ability and interests by developing artifacts and tools that use different methods of communication and/or appeal to different users. For example, students may consider using both speech and text when they wish to convey information in a game. They may also wish to vary the types of programs they create, knowing that not everyone shares their own tastes. IC.1B.2A

3 **Seek diverse perspectives for the purpose of improving computational artifacts. [SOCIAL INTERACTIONS] (P1.1) Computing provides the possibility for collaboration and sharing of ideas and allows the benefit of diverse perspectives.** IC.1B.3

a Students will collaborate and receive feedback from others. For example, students could seek feedback from other groups in their class or students at another grade level. Or, with guidance from their teacher, they could use video conferencing tools or other online collaborative spaces, such as blogs, wikis, forums, or website comments, to gather feedback from individuals and groups about programming projects. IC.1B.3A

4 **Use public domain or creative commons media and refrain from copying or using material created by others without permission. [SAFETY, LAW, & ETHICS] (P7.3) Ethical complications arise from the opportunities provided by computing. The ease of sending and receiving copies of media on the Internet, such as video, photos, and music, creates the opportunity for unauthorized use, such as online piracy, and disregard of copyrights.** IC.1B.4

a Students should consider the licenses on computational artifacts that they wish to use. For example, the license on a downloaded image or audio file may have restrictions that prohibit modification, require attribution, or prohibit use entirely. IC.1B.4A